

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Deniss Vakilov 179048IAAB

Veebipõhine vahend GNU/Linux rakenduste paigaldusskripti loomiseks

Bakalaureusetöö

Juhendaja: Edmund Laugasson

MSc

Tallinn 2020

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Deniss Vakilov

03.11.2020

Annotatsioon

Käesoleva bakalaureusetöö eesmärgiks on luua töötav avalik veebirakendus GNU/Linux'i kasutajatele, kes on puutunud kokku probleemidega pakettide paigaldusprotsessi ajal ebapiisava või puuduva kogemuse ja paigaldusprotsessi keerulisuse tõttu.

Lõputöö käigus loodud rakendus täidab eksisteerivate lahenduste puudusi. Antud rakendus on avatud lähtekoodiga ning võimaldab teistele samataoliste kogukondade liikmetele teha täiendusi ning panustada selle edasi arendamisse.

Arendusprotsessi käigus loodi veebirakendus, mis võimaldab kasutajatel otsida GNU/Linux'i distributsioonide pakette nime järgi ning salvestada leitud tulemusi „korvi“. Kasutaja soovil on võimalik genereerida salvestatud pakettidest paigaldamise skripti. Skripti saab üle vaadata veebilehel, käivitada pakutatud käsuga või alla laadida kasutaja seadmele. Rakenduse arendus on jaotatud mitmeks osadeks, kus käsitletakse veebiteenuse ja klientrakenduse ehitamist ning mõlema serverisse paigaldust.

Arendusprotsessi tulemuseks on töötav veebirakendus.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 27 leheküljel, 9 peatükki, 5 joonist, 3 tabelit.

Abstract

A Web-based Tool for Creating GNU/Linux Applications Installation Script

The aim of this bachelor's thesis is to create a working public web application for GNU / Linux users who encounter problems during the installation process, whether due to experience or complexity.

This application fills the gaps in existing solutions. The app is open source and encourages other members of the public source community to contribute.

During the development process, a web application is created that allows users to search for packages of GNU / Linux distributions either by name or by file name within the package and save the search results to a "basket". From the stored packages, at the user's request, it is possible to generate a package installation script that can be viewed on a website, run with the provided command, or downloaded to the user's device. The development is divided into several parts, which concern the construction of a web service and a client application, and the installation on both servers.

The result of the development process is a working web application.

The thesis is in estonian and contains 27 pages of text, 9 chapters, 5 figures, 3 tables.

Lühendite ja mõistete sõnastik

Arch Linux	Linux'i distributsioon
<i>checkbox</i>	Ruut, mis on märgitud linnukesega, kui on aktiveeritud
CI/CD	<i>Continuous integration/continuous delivery</i> , DevOps'i meetodika
CLI	<i>Command -line interface</i> , käsuriida
CSS	<i>Cascading Style Sheets</i> , kaskaadlaadistik
Debian	Linux'i distributsioon
DevOps	<i>Development/operations</i> , tarkvaraarenduse kultuur
Distrowatch	Veebileht Linux'i andmete statistikatega
GitHub	Lähtekoodi hostimisteenus
GNU/Linux	Linux
GUI	<i>Graphical user interface</i> , graafiline kasutajaliides
HTML	<i>HyperText Markup Language</i> , hüpertexti märgistuskeel
HTTP	<i>Hypertext Transfer Protocol</i> , hüpertexti edastusprotokoll
IP	<i>Internet Protocol address</i> , internetiaadress
IT	<i>Information technology</i> , infotehnoloogia
JSON	<i>JavaScript Object Notation</i> , lihtsustatud andmevahetusvorming
NodeJS	JavaScript'i käitussüsteem
NPM	<i>Node Package Manager</i> , JavaScript'i paketi haldur
pakettid	GNU/Linux'i kogukonnas programmid
<i>shell</i>	Käsurea interpreter
Slackware Linux	Linux'i distributsioon
SQL	<i>Structured Query Language</i> , andmebaasi päringukeel
SQL injektsioon	SQL koodi sisestamise tehnika
Suhelda domeeni kaudu	Kasutada domeeni nime (IP asemel), et saata päring sihtkohta
TCP	<i>Transmission Control Protocol</i> , edastusohje protokoll
Ubuntu Linux	Linux'i distributsioon

UNIX	Arvuti operatsioonisüsteem
URI	<i>Uniform Resource Identifier</i> , ühtne ressursiidentifikaator
URL	<i>Uniform Resource Locator</i> , internetiaadress
Veebiteenus kuulab	Veebiteenus ootab päringut

Sisukord

Autorideklaratsioon	2
Annotatsioon.....	3
Abstract A Web-based Tool for Creating GNU/Linux Applications Installation Script .	4
Lühendite ja mõistete sõnastik	5
Sisukord.....	7
Jooniste loetelu	9
Tabelite loetelu	10
1 Sissejuhatus	11
2 Hetkeolukord	12
2.1 Pakettide nimede leidmine.....	12
2.2 Pakettide paigaldamine.....	13
2.3 GNU/Linux'i distributsiooni perekonna valik.....	13
2.3.1 Distributsioonide perekondade populaarsuse võrdlus	14
2.4 Eksisteerivad lahendused.....	16
2.5 Nõuded lahendusele.....	17
3 Valdavad probleemid.....	18
3.1 Küsimustiku analüüs.....	18
4 Ülesande püstitus	23
5 Metoodika.....	24
6 Debian'i perekonna distributsioonide varamute koostöö paketi halduriga	25
6.1 Debian'i perekonna distributsioonide varamute ülesehitus.....	25
6.2 Paketihalduri konfiguratsioon.....	26
7 Tulemused	27
7.1 Veebirakenduse arenduse protsess	27
7.1.1 Kasutatud tehnoloogiate valik	27
7.1.2 Veebiteenuse arendamine	29
7.1.3 Andmete hoidmine ja töötlus.....	31
7.1.4 Paigaldamiskripti genereerimine.....	31
7.1.5 Paigaldusskripti testimine.....	32
7.1.6 Klientrakenduse arendamine	33
7.1.7 Juurutamine	34
8 Analüüs ja kasutajate tagasiside	35

8.1 Esitatud tingimuste analüüs	35
8.2 Kasutajate tagasiside.....	36
8.3 Tulevikuarendused.....	37
9 Kokkuvõte	38
Kasutatud kirjandus	39
Lisad	41
Lisa 1. Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	41
Lisa 2. Küsimustiku graafikud.....	42
Lisa 3. Veebiteenuse väljundid.....	44
Lisa 4. Klientrakenduse vaatet.....	47
Lisa 5. Korduvküsimustiku graafikud	48

Jooniste loetelu

Joonis 1. GNU/Linux'i distributsiooni ajaline järjestus [1]	14
Joonis 2. Milliste alltoodud probleemidega olete kokku puutunud?	20
Joonis 3. Kuidas leiate paigaldamisevad paketid?.....	21
Joonis 4. Kuidas paigaldate paketid?.....	22
Joonis 5. Skripti toimumise skeem	32

Tabelite loetelu

Tabel 1. Distributsioonide keskmine hinnang	15
Tabel 2. Külastuse arv iga distributsioonide perekonna kohta 12 kuu jooksul	16
Tabel 3. Külastuse keskmine arv iga distributsiooni kohta 12 kuu jooksul	16

1 Sissejuhatus

Tänapäeval avatud lähtekoodiga kogukond kasvab üsna kiiresti ning selle üheks olulisemaks osaks on GNU/Linux. GNU/Linux'ite kasutajaid tohib määratleda suure kogukonna hulka, kus igaüks saab aidata üksteist koodi kirjutamisel, koodi testimisel, vea leidmisel ning jagada loodud programmi teiste kasutajatega. Mainitud operatsioonisüsteemi eeliseks saab pidada avalikust ja vabadust programmeerimise käigus. Neid põhiprintsiipe jälgib iga avatud lähtekoodiga kogukonna liige. Eeltoodud eelised võimaldavad luua üle saja tuhandeid programme, mis on nüüd saadaval iga GNU/Linux'i kasutajale. Iga programmi jaoks saab teha mitu kloonit, mis võivad sisaldada täiendusi või parandusi. GNU/Linux'i kogukonnas programme nimetakse pakettideks. Selleks, et laadida ja edaspidi paigaldada vajalike pakette, saab kasutada distributsioonides olevatest „karbist“ soovitud pakette. Mõned need pakettidest võivad aidata kasutajatele manipuleerida teiste pakettidega ehk laadida neid alla, paigaldada, seadistada, kustutada ja teha muid operatsioone. Niisugusi pakette nimetakse paketti halduriks. Pakettide halduri kasutamiseks on vaja teada, milliste käskudega tuleb tegeleda. Selleks, et lihtsustada kirjutamise protsessi, on võimalik kasutada skripte, mis loovad graafilise liidesega programmi. Skriptilooja poolt tehtud programm peaks olema käivitav ja kättesaadav erinevatel seadmetel sõltumatu operatsioonisüsteemidest.

Bakalaureusetöö eesmärgiks on luua lahendust, mis lihtsustaks programmide otsimist ja paigaldamist GNU/Linux'i distributsioonides. Töö teoreetilises osas uuritakse, missuguste probleemidega on GNU/Linux'i kasutajad kokku puutunud pakettide otsimisel ja paigaldamisel ning hinnatakse probleemide olemusolu ja levikut. Viiakse läbi küsitlus Tallinna Tehnikaülikooli tudengite ning Kühne+Nagel ettevõtte töötajate hulgas, eesmärgiga uurida GNU/Linux'i kasutajate kogemusi programmide otsimisel ja paigaldamisel. Samuti vaadeldakse varamute ülesehitusi ning analüüsitakse võimalusi nendest vajaliku infot saamiseks. Töö praktilises osas kirjeldatakse prototüübi arendamise protsesse ehk tehnoloogiate valikut ja andmete hoidmist. Autor valis teema oma huvidest avatud lähtekoodiga kogukonda ja varamute ülesehituste tõttu ning soovil luua rakendust, mis aitab uustulnukaid alustada GNU/Linux'i kasutamisel.

2 Hetkeolukord

Tänapäeval Linux saab rohkem kasutust nii IT-valdkonna kasutajate hulgas kui ka teiste tegevusalade seas. Linux operatsioonisüsteem pakub vaba valikut nõutavate tarkvarade ja pakettide allalaadimiseks ja paigaldamiseks. See võimaldab lahendada ülesandeid palju kiiremini, mis on väga tähtis progressiivse ja eduka töö jaoks. Näiteks ühe paketi halduri jaoks saab kirjutada käsku, kus saab loetleda endale vajalike pakette ja installeerida neid oma seadmesse. Ka enamik programme saab paigaldada sisseehitatud rakenduskeskuse kaudu või terminali kaudu distributsiooni varamutest.

2.1 Pakettide nimede leidmine

Tarkvarapakettide leidmine Linux'is on ajakulukas, eriti algajatele või puuduva kogemustega inimestele. Enamikel juhtudel paketi nimi on sarnane kasutatava programmi nimega. Üks pakett võib sisaldada mitu programme, kus programmide nimed eristuvad paketi nime omast.

Üheks niisuguseks näideks võib esile tuua paketti „iputils-ping“ ja tema programmi „ping“. Eelmainitud programm on tuntud selle tõttu, et seda ei ole võimalik paigaldada programmi nime kaudu.

Selleks, et leida paketi nimi, on võimalik kasutada erinevaid lahendusvõimalusi.

Need on:

- pakettide halduri otsingu abil ehk lahendus A
- veebiotsingu mootori kaudu ehk lahendus B
- distributsiooni varamus ametlik otsing ehk lahendus C

A lahenduse korral on tarvis teada, kuidas käsutada kasutakse ehk selle tööprintsüübid. B lahenduse korral on vaja oskust päringute koostamises. Ka sellist meetodit kasutades, tuleb analüüsida ja filtreerida palju tulemusi. C lahenduse nõrgaks küljeks saab pidada seda, et mitte kõikidel distributsioonidel ja muu varamutel on olemas otsingu võimalus.

Kui töö ajal tekib olukord, kus on vaja paigaldada palju pakette, siis eeltoodud meetodid ei sobi kasutuseks. Põhjuseks on see, et kõik paketid tuleb kontrollida ühe kaupa ning see võtab palju aega.

2.2 Pakettide paigaldamine

IT spetsialistide töö hõlmab endas programmidega tegutsemist. On juhtumeid, millal süsteemi administraator vajab oma töö tegemiseks samade programmide kasutamist erinevatel seadmetel. Ka ühel ja samal seadmel võivad olla mitu operatsioonisüsteemi korraga virtuaalsete masinate ja konteinerite kasutamise tõttu. Operatsioonisüsteemide seadistamiseks kirjutakse skripte, mis omakorda võimaldavad installeerida kõik vajalikud paketid ühe käsuga.

Vaatamata sellele, et skript tundub tõhusaks lahenduseks, hõlmab see endas ka puudust. Selle meetodi nõrkuseks võib nimetada seda, et skript peab olema sobilik iga distributsiooni jaoks.

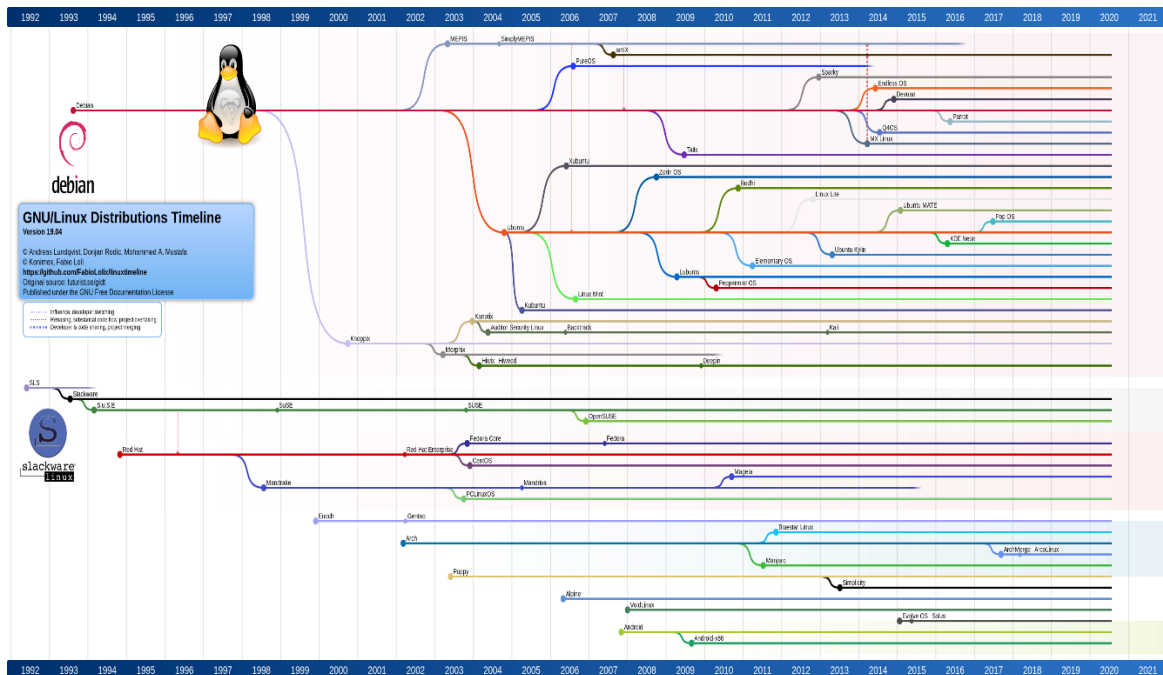
Veel üheks meetodiks pakettide paigaldamiseks on käskude kirjutamine. Tavaliselt puutuvad inimesed kokku sellise olukorraga, mille juures tuleb kirjutada pikad käsud. Ka siin tekib probleem, sest pikkade käskude käsitsi kirjutamine ei ole eelistatav. Pikka käsu tõttu vigade tekkimise oht suureneb ning tekkinud vead võivad oluliselt mõjutada kogu süsteemi, kuna paketiinstallimise käskude täitmiseks on vajalik administraatori õiguste olemasolu.

2.3 GNU/Linux'i distributsiooni perekonna valik

GNU/Linux'i põhjal on loodud tarkvara kogumik ning nimetatud distributsiooniks. Distributsioon koosneb erinevatest osadest: Linux'i tuumast, mis kuulub distributsiooni põhiosaks, GNU tööriistadest ja tekidest, lisaprogrammidest. Lisaprogrammide hulka kuuluvad aknahaldur, dokumentatsioon, töölauda keskkond ja muud.

Enamik distributsioonidest kasutavad sama litsentsi, mis GNU/Linux'il. Selle tõttu oli loodud palju tuletisdistributsioone, mille aluseks said teised distributsioonid ehk põhidistributsioonid. Tuletisdistributsioonide töötamisega saab võtta kasutuseks põhidistributsioonide ülesehitust, lisades või korrigeerides seda oma eesmärkide jaoks.

Tuletisdistributsioone ja põhidistributsioone koondab omavahel pakettide haldamise süsteem. Sellist seost nimetatakse distributsiooni “perekondadeks”.



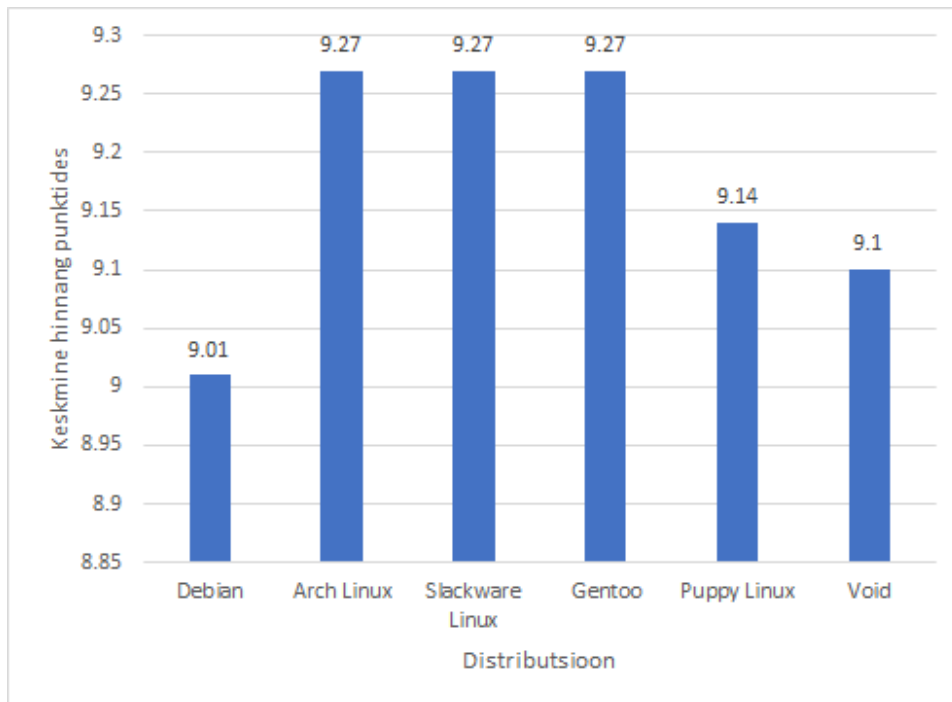
Joonis 1. GNU/Linux'i distributsiooni ajaline järjestus [1]

2.3.1 Distributsioonide perekondade populaarsuse võrdlus

Veebipõhine infoteenus distrowatch kogub andmeid külastajate kohta järgmise seisuga: see arvestatab ainult üht punkti külastatud distributsiooni veebilehele ühe IP-aadressi kohta [2] [3] [4].

Võrdlemise jaoks võeti igast distrowatch'i allikast esikümme distributsiooni.

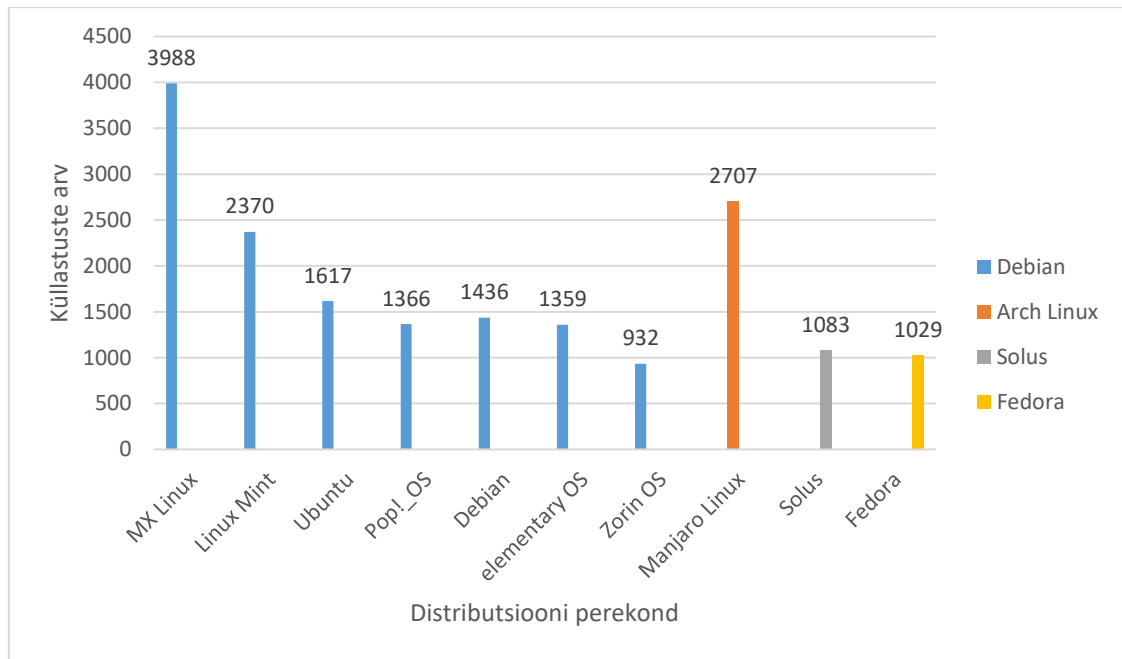
„DistroWatch Project Ranking“ tabeli andmeid määravad kasutajate hinnangud antud distributsioonide kohta [4]. Lähtudes kogutud hinnangutest, võib väita, et kõige eelistatavamad distributsioonid on Arch Linux- ja Slackware Linux'i põhised distributsioonid. Nende keskmised hinnangud on 9.27 punkti (Tabel 1). Kõige madalam hinnang on Debian-põhisel distributsioonidel, vaatamata sellele, et tulemuste vahe võrreldes Arch Linux'i ja Slackware Linux'i on ainult 0.26 punkti .



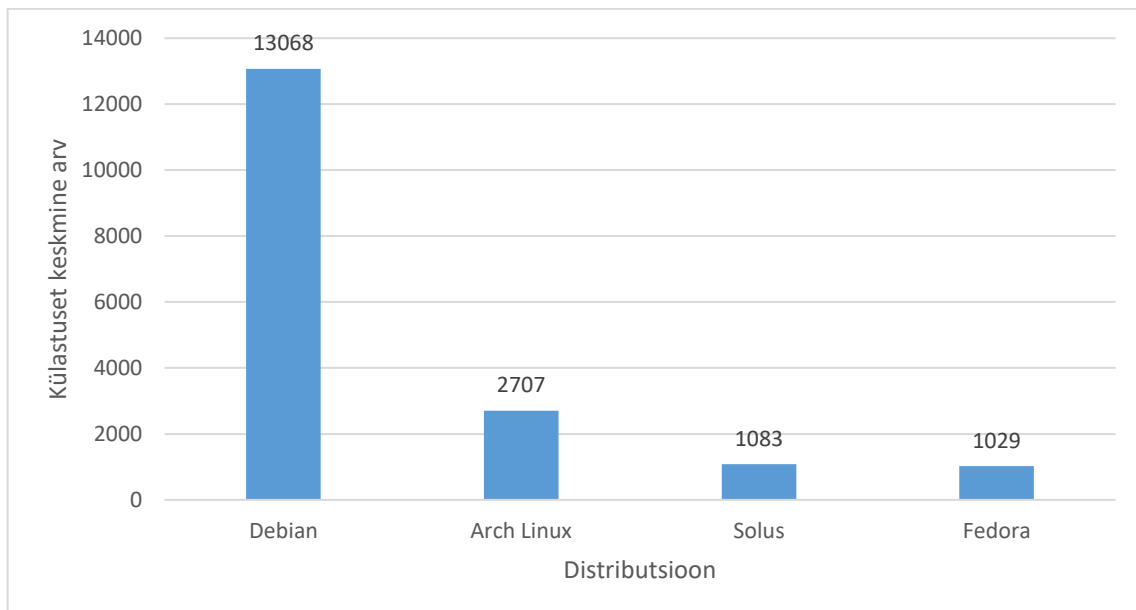
Tabel 1. Distributsioonide keskmine hinnang

“DistroWatch Page Hit Ranking” näitab, kui palju külastajaid oli distributsiooni infolehel 12, 6, 3 ja 1 kuu jooksul. Tabel 2 näitab külastajate arvu 12 kuu tagant. Statistikat võib aga viia segadusse uus distributsioon. Uue distributsiooni redaktsioon võib suruda endale rohkelt külastajate lühiajalist tähelepanu. Selline olukord võib moodustada arusaamatusi ja valet ettekujutust infolehe külastamisest.

Grupeerides distributsioonid nende perekonna järgi ja summeerides kokku külastuse arvu, oli leitud, et Debian’i perekonna kuuluvad distributsioonid on kõige külastavamaid. Võrreldes Arch Linux’i perekonnaga, keskmine külastuste vahe on rohkem kui 10,000(Tabel 3).



Tabel 2. Külastuse arv iga distributsioonide perekonna kohta 12 kuu jooksul



Tabel 3. Külastuse keskmine arv iga distributsiooni kohta 12 kuu jooksul

2.4 Eksisteerivad lahendused

Veebis on võimalik leida erinevaid lahendusi, mis katavad üks või rohkem nõudeid, kuid mitte ükski ei suuda tagada kõike.

Kasutades veebiteenust „alternativeto.net“ ning Google ja DuckDuckGo otsingu mootoreid märksõnade järgi „script“, „generator“, „linux“ ja „package“ erinevates kombitsioonides. Tulemused olid üle vaadatud, et tarkvara saaks vähemalt genereerida skripti pakettide paigaldamiseks.

Selle meetodi tulemuseks oli leitud tarkvara „Lilite“. Aga sellel tarkvaral ei ole oma pakettide otsingu aga kõvasti kodeeritud pakettide nimekirja.

Töölauarakendused on võimalik kasutada ainult antud distributsiooni perekonda jaoks, mis ei võimalda teistest perekondadest kontrollida pakette ja luua nende paigaldamiseks skripte.

2.5 Nõuded lahendusele

Iga tarkvaraga tuleb töötada talle vastava tarkvara lahendusega. Selleks, et valida sobilikku lahendust, peavad olema täidetud mitmed nõuded.

Esiteks, kasutades tarkvara lahendust, peab GNU/Linux'i kasutaja veenduma, et distributsioonide pakettide otsing ja nendest skripti genereerimine toimub ühel veebileheküljel. Teiseks peab lähtekood olema avatud ning kasutuslitsents peab võimaldama koodi kasutada. Litsents aitab vältida tootjalukustust ja tagada kasutajate lahendusest sõltumatust.

Administraator saab panna lahenduse tööle oma seadmel ja edaspidi kasutada oma ettevõtte või organisatsiooni jaoks. Lahenduse konfiguratsiooni olemasolu erinevate Debian'i perekonna varamute toetamiseks ja varamute osade välistamiseks. Tarkvara lahenduse administraator saab käsitsi uuendada varamute ja pakettide infosid või seadistada automaatse uuendust.

3 Valdavad probleemid

Linux'i operatsioonisüsteemi kasutamine võimaldab lahendada erinevaid probleeme, millega teised operatsioonisüsteemid ei saa hakkama. Näiteks programmide vabalt ümberkirjutamine ja edasiarendamine, distributsioonide lihtne paigaldamine, Linux ei nõua käivitamiseks palju ressursse ning pakub töötamiseks mugava käsurea.

Kuigi tunduvad eeltoodud Linux'i eelise näited tugevateks põhjusteks valida just selle operatsioonisüsteemi, ei piisa nendest, et enamik inimestest eelistaksid seda kasutusturul. Windows'i kui lauaarvutite operatsioonisüsteemi kasutajate osakaal siiski ületab Linux'i süsteemi kasutajaid. Põhjuseks võib pidada seda, et nii Linux'i algajad kasutajad kui ka mõned pikaajalise kogemusega kasutajad pidavad tarkvarapakettide leidmist ajakuulukaks ja keeruliseks. Selleks, et paketi haldur leiaks vajaliku paketti ja laadiks seda alla, on vaja kirjutada käsku. Samas pakette saab olla nii palju kui võimalik ning nende suure hulga tõttu muutub käsk üsna pikaks ja risk teha vigu suureneb. Lisaks sellele, ei ole mõnedel distributsioonidel graafilist liidest paketi halduri jaoks ning ainus võimalus seda kasutada on käsurea kaudu. Selline lahendus võib tekitada probleeme ja segadusi edaspidises töös, mille tõttu võib inimesel kaotada soov kasutada GNU/Linux'i distributsiooni või sundida üle minema teisele operatsioonisüsteemile.

Veebipõhine liides, mis võimaldab valida üksikuid pakette ja paigaldada need mugavalt ning kiirelt võib lihtsustada GNU/Linux'i operatsioonisüsteemi kasutamist ning tekitada rohkem huvi uutele kasutajatele.

3.1 Küsimustiku analüüs

Antud töö käigus oli läbi viidud küsimustik, mille eesmärgiks oli tõestada lõputöö probleemi aktuaalsust. Küsitlus oli suunatud ainult GNU/Linux'i kasutajatele ning selles osalesid vabatahtlikud inimesed. Küsimustikus on kokku 10 küsimust. Põhiküsimust on 9 ning need on seotud pakettide (töölauarakenduste) paigaldamise probleemidega, selle põhjustega ning ka käskude keerulise kirjutamisega. Lõpus on lisatud 1 lisaküsimus ning on suunatud nendele, kes soovivad kontrollida lõputöö käigus loodud liides. Põhiküsimuste vastuseid analüüsiti ja tehti järeldusi.

Küsimustik oli läbi viidud elektroonilises vormis ehk Google Forms kaudu. Et saada täpsema ettekujutust lõputöös tehtud liide aktuaalsusest, oli tehtud 3 küsimustiku eesti,

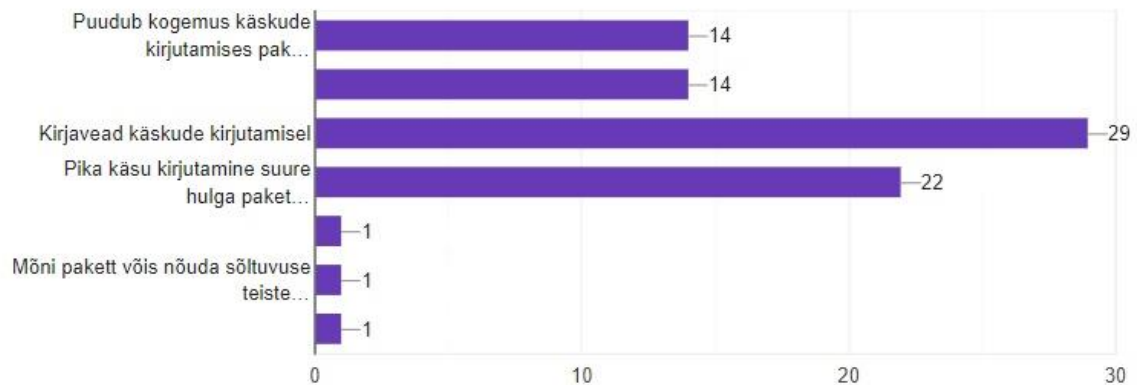
inglise ja vene keeltes. Küsitlust edastati veebipõhisel teel ning selleks kasutati Facebook'i, Instagram'i ja Zoom'i suhtlusvõrgustikke. Kokku osales 54 respondenti. Eestikeelsele küsimustikule vastas 9 inimest, mis on 16,7% kogu vastajatest. Inglisekeelsele küsitlusele vastas 24 inimest ehk 44,4% ning ülejäänud 38,9% ehk 21 küsimustiku osalejaid vastasid vene keeles. Lisas 2 on toodud küsimustiku vastused. Vastuste põhjal saab väita, et 82,5% (Lisa 2, Joonis 6) respondentidest on seotud IT-valdkonnaga ning 60,0% kasutab Linuxit õppimise jaoks. Vastustest selgus, et 57,5% vajavad eeltoodud operatsiooni süsteemi peale õppimist veel töö jaoks (Lisa 2, Joonis 7). Ka inseneride hulgas leiab GNU/Linux'i kasutajaid ja neid on 25%.

Küsimustikus selgus, et enamik ehk 42,5% vastajatest kasutab Linuxit 1-3 aastat. 10% kõikidest vastajatest ainult tutvub selle tarkvaraga ehk nende kasutamise kogemus on vähem kui 1 aasta. Suure kogemusega Linux'i kasutajaid, kes on puutunud kokku tarkvaraga 5 ja rohkem aastat, on 22,5%. On näha, et algajate ja kogunud inimeste hulk on sarnane. Neid, kes kasutab tarkvara 3-5 aastat on 25% (Lisa 2, Joonis 8).

Kuna küsimustik oli suunatud GNU/Linux'i operatsiooni süsteemi kasutajatele, enamik respondentidest vastasid, et nad kasutavad selle tarkvara regulaarselt ehk mitte vähem kui 4 tundi päevas, 15 inimest aga mitte (Lisa 2, Joonis 9). Üheks põhjuseks saaks olla see, et tarkvarapakette paigaldamine on keeruline. Et veenduda selles, oli esitatud järgmine küsimus „Kas Teil on olnud probleeme pakettide paigaldamisel?“ Toetudes vastustele, saab väita, et pakettide paigaldamine tundub keeruliseks nii algajatele kui ka teistele Linux'i kasutajatele. 43 inimest puutus kokku probleemidega pakettide paigaldamisel ning see on 80% osalejatest (Lisa 2, Joonis 10).

Küsimusele „Milliste alltoodud probleemidega olete kokku puutunud pakettide paigaldamisel?“ 54% ehk kokku 29 inimest väitis, et kõige sagedam on nad kokku puutunud kirjavigadega käskude kirjutamisel (Joonis 2).

Kui jah, siis milliste alltoodud probleemidega olete kokku puutunud: (Valige üks või mitu)

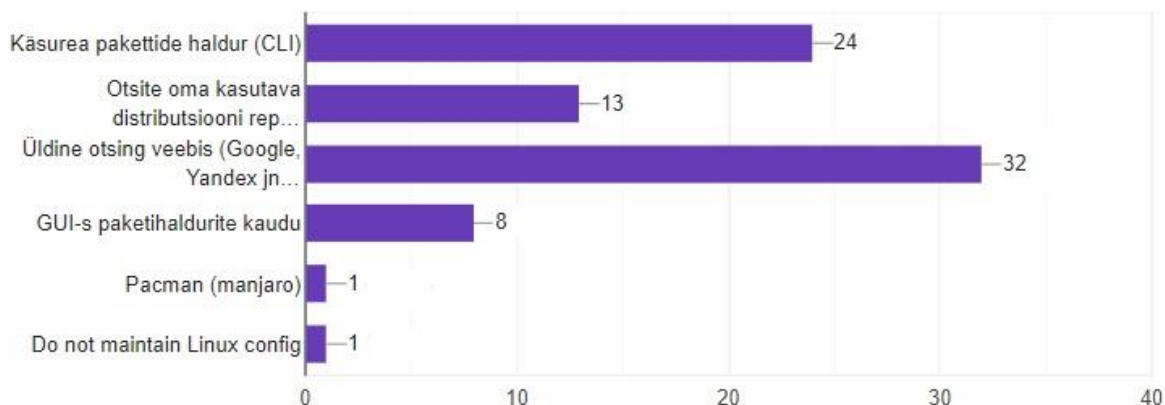


Joonis 2. Milliste alltoodud probleemidega olete kokku puutunud?

14 vastajat ei teadnud pakettide nime käskude kasutamisel. Ka üheks sagedaks probleemiks saab nimetada puuduva kogemust pakettide paigaldamiseks vajalike käskude kirjutamisel. 14 inimest tundsid paigaldamise ajal sellist raskust.

Küsimuse „Kuidas leiate paigaldamiseks vajalikud paketid?“ eesmärgiks oli teada saada, mis otsitavad meetodeid kasutavad küsimustiku osalejad (Joonis 3). 24 korda valiti käsurea pakettide halduri (CLI) abil, 13 inimest eelistavad otsida paketid oma kasutava distributsiooni ametlikus veebilehel. Kõige sagedam valiti 3 meetodit ehk üldist otsingu veebis (Google, Yandex ja teised). Sellist varianti eelistavad 32 inimest. GUI's paketi haldurite meetodit valiti 8 korda. CLI, distributsiooni ametlik veebilehekülg ja GUI's paketi halduri meetodid eristuvad sellega, et allikateks kuuluvad informatsioonid varamutest. Meetod, mille juures kasutakse üldist otsingu veebis on lihtsam. Kasutades üldist otsingut, saab leida vajalike andmeid, sõltumatu otsija tehnilisest kogemusest pakettide paigaldamisel. Seda võiks pidada üheks põhjuseks, miks enamik vastajatest eelistab sellist meetodit rohkem.

Kuidas leiate paigaldamisevad paketid? (Valige üks või mitu)

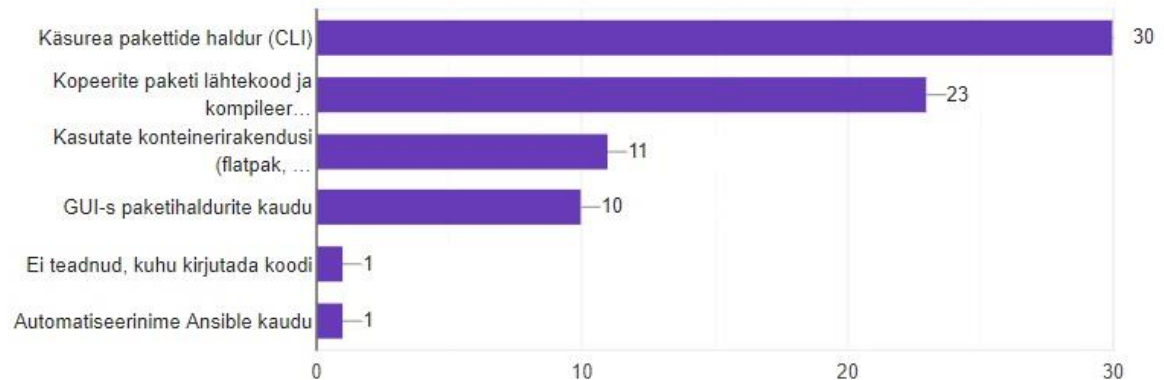


Joonis 3. Kuidas leiate paigaldamisevad paketid?

Üks põhiküsimusest „Kuidas Te paigaldate pakette?“ annab ülevaade sellest, et enamik vastajaid eelistavad poolautomatiseeritud pakettide paigaldamist. Poolautomatiseeritud lahendus annab teada, et paigaldamise protsess käis standardsel viisil. Standardisel viisil inimene ei tegutse kokku koos paigaldamise protsessiga nii palju, kui ta tegeleks sellega käsitsi. Poolautomatiseeritud meetodi kasutades tekib vähem vigu pakettide paigaldamisel. Sellise meetodi hulka saab arvama näiteks paigaldamist käsurea pakettide halduri (CLI) abil. Tulemuste järgi on näha, et 30 korda valiti sellist varianti. 11 inimest ehk 20% kogu vastajatest valisid konteinerirakenduste, näiteks *flatpak*, *snap*, *appimage*, *docker* kasutamist. GUI's paketi haldurite meetodit valiti 10 korda ehk 19% küsimustiku osalejatest (Joonis 4).

23 respondentidest eelistavad kopeerida paketti lähtekoodi ja edaspidi kompileerida oma seadmes. Sellist meetodit nimetakse käsitsi meetodiks, kus esineb palju täitmise etappe, mida on lihtne vahele jätta. Selle tõttu vigade tekkimise oht pakettide paigaldamisel suureneb. Ka käsitsi meetod nõuab programmeerimiskeele sügavama tundmist. Respondendid, kes kompileerivad paketti lähtekoodi oma seadmesse, saavad ise jälgida iga paigaldamise etapp ning muuta konfiguratsiooni. Poolautomaatne meetod ei võimalda alati ise probleemi tuvastada ja lahendada. Käsitsi meetodi abil aga saab lahendada selliseid tekkivaid vigu.

Kuidas paigaldate paketid? (Valige üks või mitu)



Joonis 4. Kuidas paigaldate paketid?

Viimase küsimuse tulemuste kaudu sai näha, et 67,5% osalejatest on huvitatud platvormist sõltumatu graafilise liidest, mis võimaldab valida vajalikud tarkvarapaketid ja luua nende paigaldamiseks sobiv skript (Lisa 2, Joonis 11). Küsimustiku lõpus oli võimalus respondentidel jätta oma kontakte, et saada programmi prototüüpi testimiseks. Liidest sooviti testida 12 inimest.

Küsimustiku eesmärgiks oli tõestada, et GNU/Linux'i kasutajad tunnevad raskusi pakettide paigaldamisel. 80% inimestest kinnitasid, et neil tekkisid arusaamatused. Kuna käskude kirjutamine võib GNU/Linux'i tarkvaral esineda ka mitme realises kujus, oodatavasti sai sagedamaks probleemiks käskude kirjutamisel tekkivad kirjavead.

4 Ülesande püstitus

Töö põhiülesandeks oli luua veebipõhist (sõltumatu) Linux'i tarkvarapakettide valimiseks ja paigaldusskriptide loomiseks keskkonda. Eeldakse, et töö käigus välja kujundatud lahendus võimaldab valida tarkvarapakette Debian GNU/Linux'i ja Ubuntu Linux'i varamutes. Tulemuseks pidi saama lahendus ehk veebipõhine graafiline liides, mis oli keskendunud Linux'i distributsiooni tarkvarapakettide andmebaasile ning näitaks tarkvarapaketi kirjeldust ja kasutusvaldkonda.

Kui tarkvarapaketid olid valitud ja skript loodud, siis võimaldaks veebipõhine keskkond seda skripti alla laadida. Lisaks näidatakse skripti sisu, et kasutaja saaks veenduda ja uurida seda enne alla laadimist ja käivitamist.

Loodav paigaldusskript peab tagama tarkvarapakettide paigaldamise. Tõrge korral paigaldusskript peab teavitada, kas GUI või CLI liide kaudu.

5 Metoodika

Lõputöö metoodika sisaldab endas praktilist, teoreetilist ja uurimusliku osasid. Esiteks seatakse nõuded võimalikeks tarkvara lahendusteks. Siis analüüsitakse ja kaardistatakse erinevad tarkvarapakettide paigaldamise mugavad lahendused. Valitakse üks nendest lahendustest nõude järgi või luuakse täiesti uus lahendus.

Luuakse Linux'i tarkvarapakettide valimise ja paigaldusskripti loomise veebipõhine keskkond, mis on otseühenduses valitud Linux'i distributsioonide tarkvarapakettide andmebaasidega. Seejuures võimaldab loodud keskkond valida üksikuid pakette.

Liides luuakse sellisena, et oleks võimalus otsida nime järgi tarkvarapakette ning leitud tulemusi lisada paigaldusskripti koosseisu.

Testitakse loodud keskkonda ja selle poolt tekitatud paigaldusskripte ja viiakse vajadusel parandused sisse.

Võimaluse korral viiakse läbi piloottestimine Linux'iga seotud õppeainetes IT Kolledžis ja kogutud tagasiside põhjal tehakse vajadusel täiendavad parandused.

Luuakse virtuaalmasinana vajalik taristu kus loodud lahendus töötab.

Luuakse ka täpne tehniline kirjeldus, kuidas loodud lahendust ehitada ja tööle panna. Seda eesmärgiga tagada jätkusuutlikkus, et oleks võimalik tulevikus ka uute Linux'i distributsioonide põhjal sama lahendust tööle panna.

6 Debian'i perekonna distributsioonide varamute koostöö paketi halduriga

Distributsiooniks varamud on üks olulisemaid ressursse. Et tagada distributsiooni kasutajale parim valik rakendusest turul, peab varamu pakkuma laia spektrit pakettidest. Distributsiooni arendajate jaoks ei ole võimalik paigaldada kõik paketid, mis on saadaval. Isegi kui oleks võimalik, see muutuks distributsiooni suurus pörs hiiglaslikuks, segaks kasutajat pikka valikuga, koormaks süsteemi ja keerustaks pakettide uuendamise protsessi. Seepärast kasutatakse varamud, mis hoiavad iseendas kõik paketid, mida on võimalik paigaldada kasutaja süsteemile.

Debian'i tuletised kasutavad Debian'i varamu struktuuri [5]. See annab võimalust saada infot igast Debian'i põhise distributsioonist kasutades sarnast meetodit.

6.1 Debian'i perekonna distributsioonide varamute ülesehitus

Varamu sisaldab mitmeid redaktsioone. Igal redaktsioonil on oma koodinimi ja varjunimi(alias) (*stable*, *oldstable*, *testing*, *unstable*), mis viitab koodinimele. Varjunimed varieeruvad distributsioonist distributsioonile. Et lihtsustada kasutaja arusaam, milliseid litsentsimistingimusi sisaldab pakett, on jagatud iga redaktsioon komponendiks. Debian ise kasutab *main*, *contrib* ja *non-free* komponente, kuid nende nimed iga distributsioon valib ise. Iga paketi sees on tarkvara, mis võib kompileeritud või loodud individuaalselt protsessori arhitektuuri jaoks. Selle tõttu jagati paketid komponentides ka arhitektuuri järgi. Varamutest on võimalik alla laadida paketid lähtekoodiga ja kompileerida seda ise ka [6] [7].

Selle uuringu raames oli vaja teada saada ainult kolm tüüpi varamu info faili: „Release“, „Packages“ ja „Contents“.

Varamus asuv „Release“ fail sisaldab üldinfot redaktsiooni kohta. Kuid meie lahenduse jaoks väärtuslikud andmed sellest failist on „Label“, „Version“, „Codename“ ja „Date“ ridad [6] [7].

„Packages“ fail koosneb kõikidest pakettide meta-andmetest. Pakettide ridade vahel on üks tühi rida, selleks et paketi haldur saaks teada, kust uue paketi meta-andmed

alustatakse ja kus lõpetatakse. „Packages“ fail on tihendatud ja faili nimi sisaldab laiendit „.gz“ [6] [7].

Iga pakett sisaldab üks või mitu faili. Kui on teada faili nimi, siis on võimalik leida paketti nimi „Contents“ varamu failist. Igal real on pakettis sisaldava faili tee ja paketi nimi. „Contents“ faili nimi peab sisaldama „Contents-“, protsessori arhitektuuri nimi ja laiendit „.gz“. „Contents“ faili sisaldus on tihendatud [6] [7].

6.2 Paketihalduri konfiguratsioon

Enne paketihalduri paigaldamise käskude käivitamisest tuleb seadistada halduri konfiguratsioonid. Paketi haldur nõuab konfiguratsiooni failis:

- URI ehk varamu juur
- „deb“ - kompileeritud pakett või „deb-src“ – lähtekoodiga pakett
- redaktsiooni nimi
- üks või mitu komponendi nimed

Vajadusel on võimalik määrata protsessori arhitektuuri nimi.

Paketihaldur sõelub konfiguratsiooni failid ja koostab saadud andmetest URL viited. Kasutades koostatud URL viiteid, teeb paketihaldur päringuid varamu serveri poole, et saada vajaliku infot või alla laadida pakette.

Paketihalduri uuendamise käsuga laaditakse all kasutaja seadmele „Packages“ faili. Paigaldamise käsuga peavad olema edastanud paketi nimed, et paketihaldur saaks leida „Packages“ failist paketi asumise tee varamus ja alla laadida neid.

7 Tulemused

7.1 Veebirakenduse arenduse protsess

Lahenduseks valiti teha veebirakendust, kuna selline viis on piiratud ainult HTTP klienti kättesaadavusega. Veebirakendus asub aadressil *http://46.101.206.218/* ja on kättesaadav brauseriga. Otsustati jagada veebirakendust veebiteenuseks ja klientrakenduseks. Selle meetodi eelised on:

- HTTP klient võiks olla nii brauser kui ka käsurea programm
- Veebiteenuse kättesaamatuse või vigade korral klientrakendus saab näidata viga sõnumit
- Klientrakenduse kättesaamatuse või vigu korral kasutaja saab kasutada veebiteenust
- Klientrakenduse ja veebiteenuse kood asuvad erinevates varamutes, mis annab võimalust rakendada CI/CD praktikat nende varamutele eraldi. See võimaldab teha kõik CI/CD etapid paindlikumaks

7.1.1 Kasutatud tehnoloogiate valik

Tehnoloogiate valik on üks tähtsamatest etappidest rakenduse arendamisel. See on pikka aega investering, mis saab kas säilitada aega ja muud ressursse või raisata neid. Sellest valikust sõltub ka rakenduse kvaliteed ja stabiilsus ning teiste inimeste huvi projektis.

7.1.1.1 Lähtekoodi hostimisteenus

Võib arvata, et lähtekoodi hostimisteenuse valik on lihtne, kuid lähemal uurimisel selgub, et seda valikut tuleb võtma arvesse ka. Sellest valikust sõltub otseselt projekti hooldamise ja haldamise mugavus. Funktsionaalsuse mitmekesisus, kasutajate arv, teenuse stabiilsus, piirangud ja maksumus - kõik ülaltoodud võivad mängida võtmerolli, millises teenuses projekt asub.

Klientrakenduse ja veebiteenuse koodi hoidmiseks oli valitud GitHub. Kõige olulisemad aspektid autori jaoks on suur kasutajate kogukond, mille abil saaks arendada, korrigeerida ja populariseerida projekti [8].

GitHub'is oli loodud kolm erinevat varamut. Üks koosnes veebi kasutaja liide(klientrakendus) koodist [9], teine veebiteenuse koodist [10] ja kolmas juurutamise skriptist ning konfiguratsiooni failidest [11].

7.1.1.2 Veebiteenuse programmeerimise keel

Veebiteenuse programmeerimise keelest sõltub nii teekide arv, mida on võimalik lisada projekti kui ka kogukonna toetus, stabiilsus, dokumentatsioonide- ja artiklite arv ning koodi kirjutamise kiirus.

NodeJS on avatud lähtekoodiga serveri platvorm Google V8 JavaScript'i mootori baasil. Paketi haldamissüsteemis NPM on rohkem kui 1,000,000 paketti, mille hulgas saab leida igasuguseid variante probleemi lahendamiseks [12].

7.1.1.3 Klientrakenduse programmeerimise keel

Klientrakendus peab olema mugav, kiire, stiilne ja arusaadav kasutajate jaoks. Selleks et täita kõik klientrakenduste arendajate nõuded, loodi graafilise liide raamistikud ja teegid. Veebiliidese jaoks eksisteerib 3 kõige enam kasutatud: Angular, React ja VueJS. Oma töö eesmärkide täitmiseks valis autor Angular'i raamistiku, sest selles on vaikimisi paigaldatud laialt kasutavaid teeke, hästi kirjutatud dokumentatsioon ja kasutamise kogemuse olemasolu.

7.1.1.4 Paigaldusskripti shell

Paljudes UNIX'i laadsetes operatsioonisüsteemides vaikimisi kasutatakse Bash *shell*'i. Bash on POSIX standartiga ühilduv käsuriida interpretaator, aga Bash'is on sisseehitatud funktsioonid, mis ei ole määratud POSIX standardiga. Selle pärast paigaldusskript on kirjutatud POSIX standartiga, mis võimaldab seda käivitada erinevates *shell*'ides, kaasa arvatud Bash.

Kuna ei ole teada, mis *shell*'i kasutatakse süsteemis, siis vaikimisi paigaldusskript kasutab *shebang*'is „/bin/sh“, mis on tavaliselt viite mingi konkreetse *shell*'i. Kui kasutaja ei määranud, mis käsurea interpretaator peab käivitama skripti, siis süsteem käivitab „/bin/sh“ interpretaatori kaudu.

7.1.1.5 Käivituskeskkond

Rakenduse (programmi) jaoks mängib olulist rolli käivitava keskkonna valik. Turvalisuse jaoks on väga oluline, et rakenduse eriosade jaoks oleks erinev keskkond. See väldib olukorda, kus ühe keskkonna maha kukkumine rikub kogu rakenduse. Stabiilsus tuleks tagada automaatse taaskäivitamisega juhul, kui rakenduses on paratamatu viga ning keskkonna asendamise või rakenduse osa värskendamise lihtsus ja kiirus.

Samuti on üsna raske haldama kõike rakenduse käivitamise sõltuvusi - omada uusim sõltuvuste versiooni või jääda mistahes versiooni juurde.

Selle probleemide lahendamiseks on hästi sobivad konteineri lahendused, kuna on vaja üks kord ehitada oma keskkonda minimaalsete sõltuvuste komplektiga konteineri pilti ja pärast selle alusel ehitada just see, mis sobib konkreetsetele nõuetele, lisades täiendavaid sõltuvusi. Internetis on juba jagatud mõned konteineri pilte, mida on võimalik kasutada aluseks ehk ühest konteineri pildist on võimalik kiiresti käivitada mitmeid konteinereid, mis annavad stabiilsust ja turvalisust.

Lõputöö autor valis Docker'i lahendust, sest see tehnoloogia on kasutatavam turul [13], ka laia hulga juhendite ja dokumentatsioonide olemasolu ning kogemuse tõttu.

7.1.2 Veebiteenuse arendamine

Veebiteenuse töötamiseks oli vaja andmeid varamutest. Selle jaoks tuli veebiteenuse administraatorile täita konfiguratsiooni failid „config.json“ ja „.env“.

„config.json“ failil sai seadistada kuuluva TCP porti ning andmeid varamute kohta. Varamu seadistamiseks on vaja kirjutada sisse:

- Distributsiooni nimi
- Varamu juurviide
- Varamu juurkausta
- Redaktsiooni nimede filtrid
- Komponenti nimed
- Protsessori arhitektuuri filtrid
- „Contents“ faili nimede filtrid

Kasutati kaks filtri tüüpi: „basicFilter“ - on regulaarne väljend, et filtreerida muustriga ja „excludeFilter“ - nimikiri nimede välistamiseks.

„.env“ fail koosneb keskkonna muutujatest, et paindlikult muuta kaustade teed. See fail võib sisaldada projekti juurkausta, allalaadimise kausta ja lahtipakkimise kausta teed.

Kui konfiguratsiooni failid on seadistatud, siis sai alla laadida „Packages“ ja „Contents“ failid ja koostada varamute „kaardi“. Selleks käivitati „updateRepoMap/fetchRepoData.js“ faili, mis loeb „config.json“ ja „.env“ ning koostab viiteid „Packages“ ja „Contents“ failide alla laadimiseks ja nende lahti pakkimiseks. Kui alla laadimine õnnestus, siis pandi „kaardi“ faili („repositoriesMap.json“) kasutatud konfiguratsiooni osad. See viis tagas, et konfiguratsiooni faili väljad olid määratud õigesti ja sattuvad „kaardi“ faili.

Veebiteenus kuulab 3 *endpoint*'i:

- /api/map – tagastab varamute „kaardi“, mis näitab toetavaid distributsioone, redaktsioone, arhitektuure ja komponente. Kliendi rakendus kasutab seda, et täita vormi filtrid (Lisa 3, Joonis 12)
- /api/search – kontrollib kaks kohustusliku argumenti - „distro“ andmetüüpiga string ja „packages“ andmetüüpiga massiiv. Kui olid edastanud valed argumendid või mingi nendest olid puudu, siis tagastati kasutajale veakiri ja HTTP tagastuskoodiks sai „400 Bad Request“(Lisa 3, Joonis 14). Argumendid “release” andmetüübiga string ja “arch” andmetüübiga string on ebakohustuslikud, aga tugevasti mõjutavad vastuse ootamise ajale. Kui ei olnud mingeid vigu, tagastatakse kasutajale massiiv pakettide meta-infoga (Lisa 3, Joonis 13). Kontrolliti ka „distro“, „release“ ja „arch“ välide sisu, kasutades „kaardi“ faili. Kui mingi sellest välja sisust ei olnud määratud „kaardi“ failis, siis tagastati kasutajale veateade koos HTTP tagastuskoodiga „400 Bad Request”
- /api/generate – võtab sisse sama argumendid nagu /api/search, aga tagastab installeerimise skripti, kui ei tekkinud vigu (Lisa 3, Joonis 15). Vastasel korral tagastab vigu teadet (Lisa 3, Joonis 16)

7.1.3 Andmete hoidmine ja töötlus

Andmete hoidmiseks prooviti kasutada Postgresql-i andmebaasi. Selleks, et täita või uuendada andmebaasi, oli vaja kõigepealt „Packages“ ja „Contents“ failid alla laadida ja sõeluda neid. Tulemust edastati andmebaasi serverisse. Uuendus kestab keskmiselt 7 minuti. Turvaliste päringutute teostamiseks andmebaasi poole, kasutati teeke ja autori poolt loodud funktsioone andmebaasi serveril, et vältida SQL injektsiooni.

Töö käigus tehti teist viisi andmete hoidmiseks. Selle viisi juures ei sõeluti faile kohe pärast laadimist vaid ekstraheeriti iga kasutaja päringuga ainult seda faili osa, kus pakett asus. Selle toimimiseks kirjutas antud töö autor *shell* skripti, mis otsib pakettide meta-infosid ja prindib standardväljundi. Veebiteenus käivitab selle skripti iga päringuga serveri ja sõelub väljundi NodeJS objektide massiivi. Selle viisi kasutades ei ole enam vaja raisata ressursse andmebaasiks ning sõeluda kõik „Package“ ja „Contents“ failid.

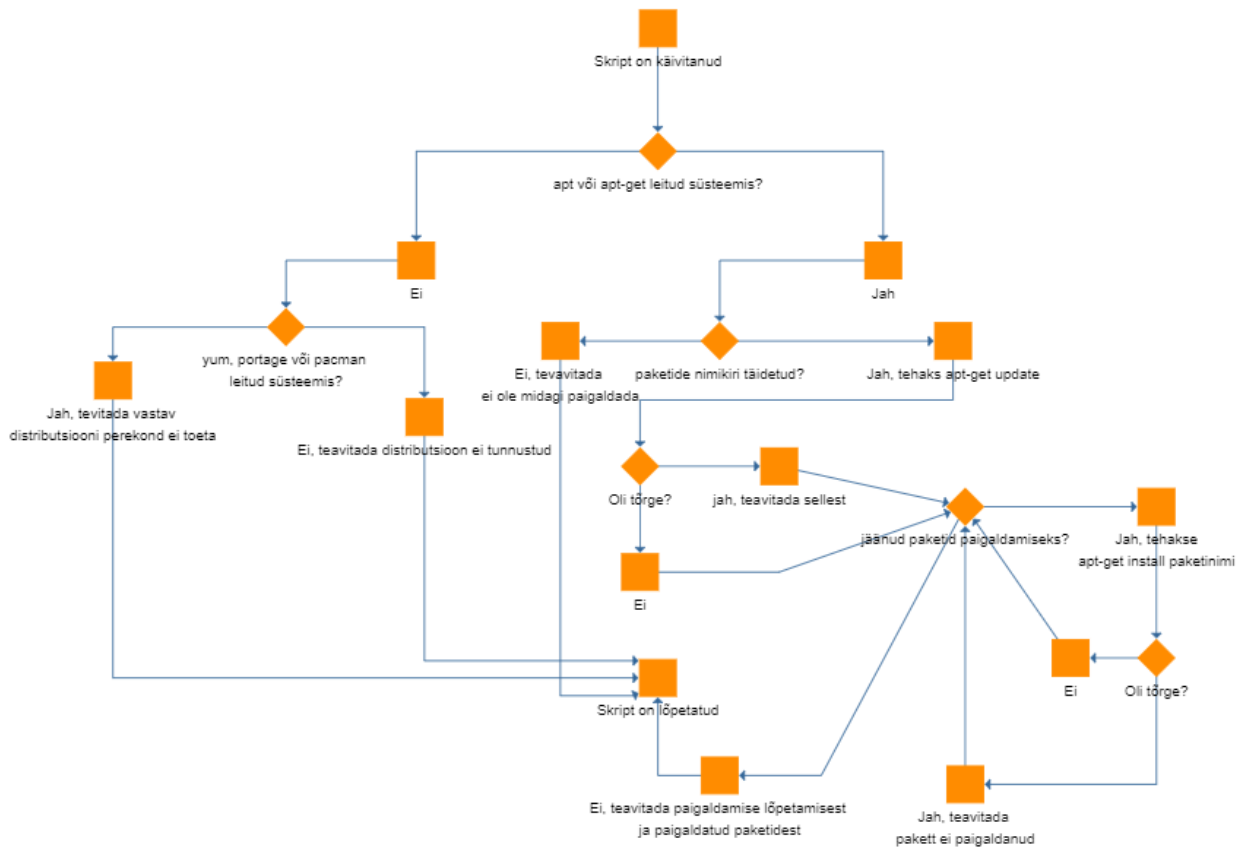
Mainitud otsingu *shell* skript võtab kaks kohustuslikku ja viis ebakohustuslikku argumente. Kohustuslikud argumendid on „distro“ ja „package“. Kui üks nendest ei ole määratud, siis näidatakse, et on tehtud viga. Ebakohustuslikud argumendid on - „rootdir“, „release“, „component“, „architecture“ ja „mode“. Argumendid „distro“, „rootdir“, „release“, „component“ ja „architecture“ on selleks, et koostada teed failisüsteemis „Package“ failideni. Argument „mode“ määrab rangust ja vaikimisi kasutatakse „strict“ väärtust, mis teeb otsingu paketi täpse nime järgi.

7.1.4 Paigaldamisskripti genereerimine

Kuna paigaldamisskriptid tulevad minimaalse erinevusega, siis on mugav moodustada malli, mille põhjal saab paindlikult lisada käsud skripti. Hetke malli sees on käsud, mis kontrollivad distributsiooni perekonda süsteemis leitud paketi halduri alusel (kui skript on käivitatud mitte Debian'i perekonda distributsioonist, siis skript lõpetab tööd ära). Käsud kontrollivad, et käivitajal oleksid „root“ kasutaja õigused, teavitavad vigadest ning annab informatsiooni paigaldamise kohta. Malli käsud aga ei luba paigaldamist, kui ei olnud edastanud mingite pakettide skripti.

Skript on kirjutatud niimoodi, et tagada edukat paigaldust ja teavitada kasutaja vigadest. Teavitamine toimud kahel erineval viisil. Skript vaatab järele, kas kasutajal on praegu

töölauda keskkond või mitte. Kui on, siis paigaldatakse „zenity“ paketti, mille kaudu on võimalik näidata teateid kasutajale. Vastasel juhul prindib skript teateid terminali.



Joonis 5. Skripti toimumise skeem

7.1.5 Paigaldusskripti testimine

Tehtud teenusel kõige olulisem väljund on stabiilselt töötav skript, mis teeb mingit otsust erinevates juhtumites. Selleks on vaja kirjutada erinevaid teste, mis kontrollivad erinevaid olukordi ehk kas tegemist on soodsa asjaseisuga või mitte.

Testide kirjutamiseks on vajalik testimiskeskonda, võib kasutada testimisraamistikku. Autor valis shUnit2 raamistikku, sest see pakett on hästi toetav ning pakub lihtset liidet tingimuste kontrollimiseks. Testimiskeskonna loomiseks kasutas autor Docker'i konteinereid, mille kaudu sai simuleerida erinevaid distributsioone.

Testimise jaoks oli kirjutatud testimiskriipt [13], mis kopeerib paigalduskriipti konteinerisse koos määratud pakettide nimekirjaga ja käivitab seda. Väljund ja tagastuskood(väljumiskood) võrreldakse koos sobiva tulemusega testimisraamistiku kaudu.

7.1.6 Klientrakenduse arendamine

Angular on raamistik, mis annab võimalust kasutada komponentide lähenemisviisi. Komponente saab kasutada mitu korda erinevates rakenduse kohtades. See vähendab koodi duplikaadi ja mahtu. Tinglikult klientrakendust jagatakse kaheks peamisteks komponentideks: genereerimise komponent ja otsingu komponent.

Otsingu komponent laaditakse päringuga „/search“ *endpoint*'i. Avamise jooksul klientrakendus teeb päringu veebiteenusele „/api/map“, et saada infot filtreerimisvormi jaoks. Vorm koosneb väljadest „Distribution“(kohustuslik väli), „Release“, „Architecture“ ja „Package name“ (kohustuslik väli). Valides mingi väärtus vormi väljas, filtreerib klientrakendus automaatselt väärtusi teistes väljades, välja arvatud „Distribution“ ja „Package name“ välju. Otsingut ei toimu enne, kui kasutaja on määranud vormis kõik kohustuslikud väljad. Kui kõik kohustuslikud väljad on määratud, siis toimub otsing automaatselt. Juhul kui tekkib mingi viga veebiteenuse pool, näidatakse kasutajale teade sellest veast. Vastuseks otsingupäringule saadab veebiteenus JSON formaadis infot pakettidest (Lisa 4, Joonis 17). Ka tühja vastuse puhul (kui ei leitud vasted), näitab klientrakendus teadet selle kohta (Lisa 4, Joonis 18). Valides sobiva paketi *checkbox*'iga tabelis ja salvestades „Add“ nupuga, lisatakse paketi info brauseri salvestusruumi.

Otsingu komponent laaditakse päringuga „/search“ *endpoint*'i. Avamise jooksul, kuvab klientrakendus tabelleid koos salvestatud pakettidega võetud brauseri salvestusruumist. Tabelid jagatakse distributsiooni nimedeks. Valides sobiva paketi *checkbox*'iga tabelis ja vajutades „Generate“ nuppu, koostakse paigalduskriipt (Lisa 4, Joonis 19). Skripti saab läbi vaadata brauseris „Show Script“ nupuga või alla laadida „Download“ nupuga. Allalaetud faili nimi vaikimisi on „install_packages.sh“. Juhul kui ei leitud ühtegi paketi infot brauseri salvestusruumis, siis näidatakse teadet, et on vaja lisada paketid, kasutades „search“ veebilehti (Lisa 4, Joonis 20). Et kustutada salvestatud pakettide infosid, on vaja valida tabelist *checkbox*'iga read ja vajutada „Delete selected“ nupu.

7.1.7 Juurutamine

Kuna rakendus koosneb kahest põhiosadest (klientrakendus ja veebiteenus), mis eksisteerivad sõltumatu üksteisest (ehk kui üks pool kukub, teine saab osaliselt või täielikult töötada) ja kood asub erinevates varamutes, siis ei ole lihtne panna neid töötada ühel serveril, kuid erinevatesse käivitus keskkondi korraga. Põhimõtteliselt, kasutades Docker'i, on vaja panna klientrakendus ja veebiteenus oma konteineri koos konfiguratsioon failidega ja seadistada kommunikatsioon üksteisega. Selle probleemi lahendamiseks oli loodud „install.sh“ *shell* skript [11], mis haldab juurutamisprotsessi automaatselt.

Selle skripti toimumiseks on vaja kuut programmi: git, ng (Angular'i käsurea programm), docker, npm ja docker-compose. Enne käivitamist kontrollib skript kõike neid programme ja lõpetab tööd, kui mingi neist on puudu. Git'i kaudu võetakse kõige uuemad versioonid klientrakenduse ja veebiteenuse varamutest. Kui git on valmis, siis ng programm interpreteerib klientrakendus TypeScript koodi JavaScript'i ja paneb kõik JavaScript -, HTML-ja CSS koodi kolme faili ühte kataloogi ja kopeerib seda „nginx.conf“ kataloogi.

Peale klientrakenduse interpreteerimist on saadud kolm faili, mida on vaja host'ida veebiserveriga. Nginx on veebiserver, mida kasutatakse laialdaselt turul ning sellel on suur kogukond kogu maailmas. Klientrakenduse ja veebiteenuse aluseks valiti kaks Docker'i pilti „Alpine Linux“ distributsiooniga koos eelpaigaldatud Nginx'i- ja NodeJS'iga.

Viimane etapp oli kopeerida koodi ja konfiguratsiooni failid Docker'i konteinerite sisse. Iga põhiosaks on oma Dockerfile. Programm docker-compose loeb „docker-compose.yml“ faili, mis kirjeldab Dockerfile'ide asukohti, taaskäivituse tõrge juhul ja domeen nimed Docker'i võrgus. Kui Docker'i pildid on ehitatud, docker-compose paneb neid ühe isoleeritud võrku, kus nad saavad suhelda domeeni kaudu. Nginx'i konfiguratsiooni failist („default.conf“) on näha, et veebiserver edastab „/api“ päringud „nodeserver“ poole. See domeen on määratud „docker-compose.yml“.

8 Analüüs ja kasutajate tagasiside

Selle bakalaureusetöö jaoks oli esitatud mitu lähtetingimust. Kõige olulisemad tingimused oli lahendada Linux'i kasutajate probleemid seotud pakettide paigaldamisega, kasutades paketihalduri otse. Esimese küsimustiku tulemuste järgi selgus, et kasutajatel on probleeme varamutes pakettide leidmisega, ebapiisav kogemus käskude kirjutamisel ning pikkade käskude kirjutamise tõttu suure hulga kirjavigade tekkimine.

Kolme peamise probleemi lahendamiseks, esitas autor mitu tingimust. Eesmärkide saavutamiseks ja pärast nende täitmiseks, korraldas korduv küsimustik.

8.1 Esitatud tingimuste analüüs

Autori poolt määratud tingimused olid täidetud osalises või täielikus määras.

Järgmised tingimused on täidetud täielikult:

- otsing ja skripti genereerimine on ühel veebilehel
- otsing nime järgi ja pakettide nimede panemine skripti
- lähtekood on avatud ja kasutuslitsents võimaldab koodi vabalt kasutada – kasutatakse GPLv2 litsents
- iga süsteemi administraator saab panna seda lahendust serveri oma organisatsiooni jaoks – kood saab vabalt võtta varamutest ja paigaldada autori poolt pakutud paigaldamisskriptiga [11]
- konfiguratsiooni fail võimaldab kasutada erinevaid Debian'i perekonna varamuid ja välistada varamute osad – konfiguratsiooni fail annab võimalust kasutada Ubuntu ja Debian'i varamud. Teoreetiliselt on võimalik lisada iga Debian'i perekonna varamu konfiguratsiooni faili
- pakettide infosid on võtnud otse varamutest – konfiguratsiooni failis on võimalik määrata varamu URL'i
- skripti saab läbi vaadata veebirakenduses
- administraatoril on võimalus uuendada varamute ja pakettide infosid kas käsitsi või automatiseerimise teel – seadistada crontab programmi, et käivitada käsu „node downloadAndParsePackageFiles.js“ või käivitada seda ise
- tõrge korral skript teavitab kasutajat GUI või CLI kaudu – skript näitab teated GUI's „Zenity“ programmi kaudu või prindib CLI'i sisse

- luuakse virtuaalmasinasse vajalik taristu, kus loodud lahendus töötab – lahendus jookseb virtuaal serveril aadressiga <http://46.101.206.218/>

8.2 Kasutajate tagasiside

Töö käigus loodud teenuse aktuaalsuse ja vajaduse tõestamist näitas korduvküsimustik. 12 inimest soovisid testida veebirakendust, kuid vastasid nendest 10. Respondendid pidid andma oma hinnangu üldise veebirakenduse kasulikkust, vastates küsimustele, mis sisaldavad nii 1-5 palli hindamissüsteemi (kus 1 – üldse ei ole rahul/nõus ja 5 – olen väga rahul/nõus) kui ka probleemide kirjeldamise vastusteid.

Pool vastajatest ehk 5 inimest hindasid pakettide paigaldamist pallile 4 (Lisa 5, Joonis 21). 3 respondendi jäid väga rahul pakettide leidmisega ning 2 vastajatest panid hindeks 3. Küsimusele „Kas Teil tekkisid probleemid pakettide leidmisel?“ vastas 3 inimest ning kõik nendest puutusid kokku aeglase pakettide otsinguga. Probleemi põhjuseks saaks olla see, et kasutaja ei määranud kõik filtrid ja pakettide leidmine toimub *shell* skripti kaudu. Ka paigaldusskripti saamisel üks vastajatest puutus kokku järgmise probleemiga – käsu sees sai jutumärk pandud vales kohas, mille tõttu käsk sai lõpetamata. Tagantjärele muutis autor nõutavas kohas kahekordset jutumärki ühekordsele, millega lahendas tekkiva probleemi.

Vastustest küsimusele „Kui rahul olete pakettide paigaldamisega, kasutades allalaaditud skripti“ selgus, et 3 inimest on väga rahul, 4 inimest ehk 40 % panid hindeks 4 (Lisa 5, Joonis 22). 1 respondentidest aga ei olnud üldse rahul. 70% vastajatest on nõus, et pakutud teenus saab olla kasulik Linux'i kasutajate algajatele, kuna see võimaldab mõista parem Linux'i tarkvara või mängida skripti koodiga. 30% või 3 inimest ei ole nõus, et teenus saab algajatele abiks (Lisa 5, Joonis 23). 6 küsimustiku vastajatest arvavad, et pakutud teenuse abil kogenud Linuxi kasutajad (näiteks süsteemide administraatorid) saavad aidata või toetuda algajaid kasutajaid. 4 inimest aga ei ole selle väitega nõus.

Viimase küsimuse eesmärgiks oli teada saada teenuse kasulikkust nii vastajate kui ka teiste Linuxi kasutajate hulgas. Küsimusele „Kas te soovitaksite seda teenust oma tuttavale/sõbrale“ 30% inimestest kindlasti soovitaksid seda, 50% vastasid, et pigem olevad nõus (Lisa 5, Joonis 24). 1 vastaja aga ei soovi seda teenust teistele.

8.3 Tulevikuarendused

Valitud laia teema tõttu pidi lõputöö autor valima üht distributsiooni perekonda. Kuna on olemas palju teisi distributsiooni perekondi, mida kasutavad suur hulk inimesi, on tarvis kohandada veebiteenust nendele distributsioonidele. Selle tõttu saab aidata Linuxi kasutajatele vältida antud töös käsitletud probleeme ja lihtsustada pakettidega tehtavat tööd.

Otsingu jõudluse paremaks tegemine. Praegune otsingulahendus on suhuliselt aeglane, kuna sõelub andmefailid ja tagastab leitud informatsiooni vastuses. Seda on võimalik kiirustada andmebaasi abil, aga on vaja mõelda, kuidas kiiresti ja mugavalt uuendada andmeid, kuna tavaliselt on varamu sees mitu erinevaid mahukaid andmefaile paketidest.

9 Kokkuvõte

Bakalaurusetöö eesmärgiks oli arendada välja veebirakendust, mis võimaldaks lahendada Linuxi kasutajate tekkivaid probleeme pakettide paigaldamise protsessi jooksul. Lõputöö tulemuseks oli loodud teenus, mis täidab probleemilahenduse nõudeid.

Töö eesmärgi saavutamiseks oli korraldatud nii praktiline, teoreetiline kui ka uuringuline töökäik. Esiteks tehti uuringulist osa, kus tõestati lõputöö probleemi aktuaalsuse olemasolu. Esimese küsimustiku käigus selgus, et enamik ehk 80% respondentidest tunnevad raskusi pakettide paigaldamisel. Tihti puuetakse kokku kirjavigadega käskude kirjutamisel ning suure hulga pakettide paigaldamise tõttu pika käsu kirjutamisega. Järgmisena analüüsiti distributsioonide statistilise informatsiooni ja valiti selle põhjal distributsiooni perekonda. Et kirjutada rakendust, mis lahendaks lõputöös valdavaid probleeme, uuris autor valitud distributsiooni perekonda varamute ülesehitust.

Praktilises osas tehti klientrakendust ja veebiteenust. Veebiteenus kogub informatsiooni varamust ja edastab pakettidest küsitud informatsiooni kliendile. Klientrakendus pakkub mugavat viisi päringu koostamiseks veebiteenuse poole. Veebiteenuse tulemuseks sai klient *shell* skripti, mis võimaldab paigaldada nõutavaid pakette.

Töö lõpus korraldati korduvküsimustiku, mille vastuste põhjal tõestati lõputöö aktuaalsust. 70% vastajatest arvavad, et loodud teenus aitab Linux'i algajatele kasutajatele lihtsustada pakettide leidmist ja paigaldamist. Ka 60% respondentidest märkasid, et tehtud veebirakendus saaks olla kasulik ka kogunud Linux'i kasutajatele.

Kasutatud kirjandus

- [1] u/philippeclercq, „Linux Distributions Timeline, but reduced to the top 50 distributions on Distrowatch and their ancestors,“ 28 07 2020. [Võrgumaterjal]. Available: <https://i.redd.it/aygzaivcbmd51.png>. [Kasutatud 28 11 2020].
- [2] A. A. Limited, „Distrowatch Page Hit Ranking,“ Atea Ataroa Limited, [Võrgumaterjal]. Available: <https://distrowatch.com/dwres.php?resource=popularity>. [Kasutatud 05 11 2020].
- [3] A. A. Limited, „Distrowatch Page Hit Ranking Trends,“ Atea Ataroa Limited, [Võrgumaterjal]. Available: <https://distrowatch.com/dwres.php?resource=trending>. [Kasutatud 05 11 2020].
- [4] A. A. Limited, „Distrowatch Project Ranking,“ Atea Ataroa Limited, [Võrgumaterjal]. Available: <https://distrowatch.com/dwres.php?resource=ranking>. [Kasutatud 05 11 2020].
- [5] Debian, „Debian derivatives,“ 14 07 2020. [Võrgumaterjal]. Available: <https://www.debian.org/derivatives/>. [Kasutatud 24 11 2020].
- [6] Debian, „DebianRepository,“ 14 11 2020. [Võrgumaterjal]. Available: <https://wiki.debian.org/DebianRepository>. [Kasutatud 15 11 2020].
- [7] Debian, „Debian Repository Format,“ 18 05 2020. [Võrgumaterjal]. Available: <https://wiki.debian.org/DebianRepository/Format>. [Kasutatud 15 11 2020].
- [8] L. Harkushko, „The List of Top Source Code Hosting Services,“ Yalantis, [Võrgumaterjal]. Available: <https://yalantis.com/blog/github-and-alternatives-what-is-the-best-hosting-service-for-your-software/>. [Kasutatud 16 11 2020].
- [9] D. Vakilov, „scripty-frontend,“ scripty-backend, 29 04 2020. [Võrgumaterjal]. Available: <https://github.com/denvaki/scripty-frontend>. [Kasutatud 03 01 2021].
- [10] D. Vakilov, „scripty-backend,“ 29 04 2020. [Võrgumaterjal]. Available: <https://github.com/denvaki/scripty-backend>. [Kasutatud 03 01 2021].
- [11] D. Vakilov, „scripty-install,“ 07 12 2020. [Võrgumaterjal]. Available: <https://github.com/denvaki/scripty-install>. [Kasutatud 03 01 2021].
- [12] S. M. Liran Tal, „npm passes the 1 millionth package milestone! What can we learn?,“ Snyk Limited, 04 06 2019. [Võrgumaterjal]. Available: <https://snyk.io/blog/npm-passes-the-1-millionth-package-milestone-what-can-we-learn/>. [Kasutatud 18 11 2020].
- [13] D. Vakilov, „test_templateOfApt.sh,“ 03 01 2021. [Võrgumaterjal]. Available: https://github.com/denvaki/scripty-backend/blob/master/testForScriptTemplate/test_templateOfApt.sh. [Kasutatud 05 01 2021].
- [14] L. S. Sterling, The Art of Agent-Oriented Modeling, London: The MIT Press, 2009.
- [15] Debian, „Derivatives,“ 27 10 2018. [Võrgumaterjal]. Available: <https://wiki.debian.org/Derivatives>. [Kasutatud 22 11 2020].
- [16] Debian, „Supported Hardware,“ Debian, [Võrgumaterjal]. Available: <https://www.debian.org/releases/stable/i386/ch02s01.en.html#idm185>. [Kasutatud 24 11 2020].

- [17] instructure.com, „Distribution Families,“ [Võrgumaterjal]. Available: https://canvas.instructure.com/files/37952368/download?download_frd=1. [Kasutatud 16 11 2020].
- [18] Atea Ataroa Limited, „Fedora,“ [Võrgumaterjal]. Available: <https://distrowatch.com/table.php?distribution=fedora>. [Kasutatud 15 11 2020].
- [19] Red Hat, Inc. and others, „Download Fedora 33 Workstation,“ [Võrgumaterjal]. Available: <https://getfedora.org/en/workstation/download/>. [Kasutatud 15 11 2020].
- [20] Atea Ataroa Limited, „Search Distributions, Based on Debian, Status Active,“ [Võrgumaterjal]. Available: <https://distrowatch.com/search.php?ostype=All&category=All&origin=All&basedon=Debian¬basedon=None&desktop=All&architecture=All&package=All&rolling=All&isosize=All&netinstall=All&language=All&defaultinit=All&status=Active#simple>. [Kasutatud 15 11 2020].
- [21] Atea Ataroa Limited, „Search Distributions, Based on Fedora, Status All,“ [Võrgumaterjal]. Available: <https://distrowatch.com/search.php?ostype=All&category=All&origin=All&basedon=Fedora¬basedon=None&desktop=All&architecture=All&package=All&rolling=All&isosize=All&netinstall=All&language=All&defaultinit=All&status=All#simple>. [Kasutatud 15 11 2020].
- [22] Atea Ataroa Limited, „Search Distributions, Based on Fedora, Status Active,“ [Võrgumaterjal]. Available: <https://distrowatch.com/search.php?ostype=All&category=All&origin=All&basedon=Fedora¬basedon=None&desktop=All&architecture=All&package=All&rolling=All&isosize=All&netinstall=All&language=All&defaultinit=All&status=Active#simple>. [Kasutatud 15 11 2020].
- [23] Atea Ataroa Limited, „Search Distributions, Based on Debian, Status All,“ [Võrgumaterjal]. Available: <https://distrowatch.com/search.php?ostype=All&category=All&origin=All&basedon=Debian¬basedon=None&desktop=All&architecture=All&package=All&rolling=All&isosize=All&netinstall=All&language=All&defaultinit=All&status=All#simple>. [Kasutatud 15 11 2020].

Lisad

Lisa 1. Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Deniss Vakilov

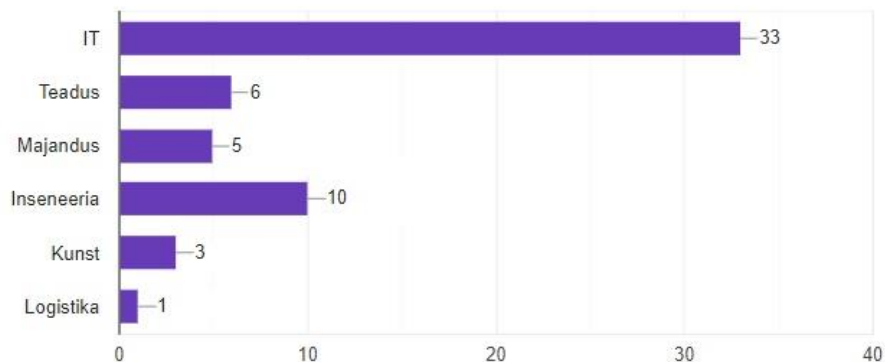
1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Veebipõhine vahend GNU/Linux rakenduste paigaldusskripti loomiseks“, mille juhendaja on Edmund Laugasson
 - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

07.01.2021

¹ Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

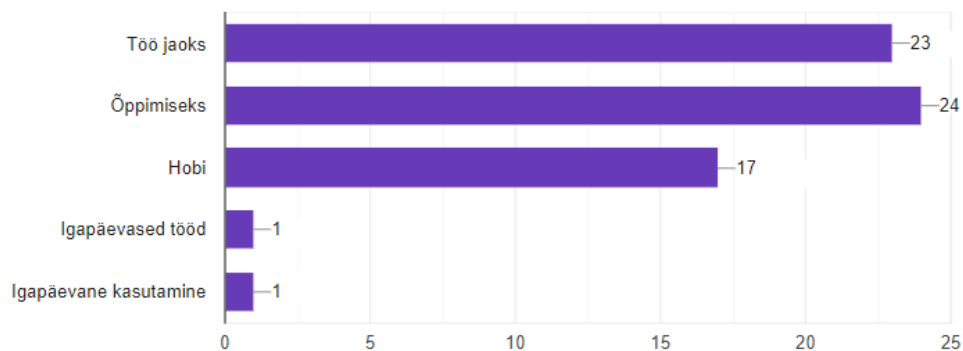
Lisa 2. Küsimustiku graafikud

Mis valdkonnas Te töötate / õpite? (Valige üks või mitu)



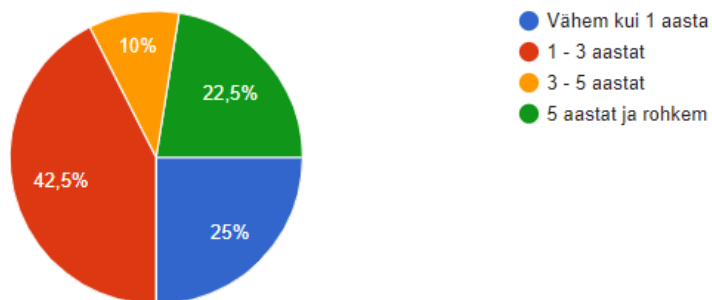
Joonis 6. Mis valdkonnas Te töötate/õpite?

Mis eesmärgil te kasutate Linuxi? (Valige üks või mitu)



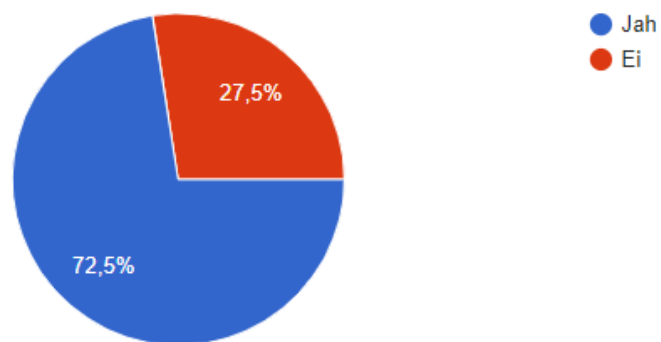
Joonis 7. Mis eesmärgil te kasutate Linuxi?

Kui kaua olete Linuxit kasutanud?



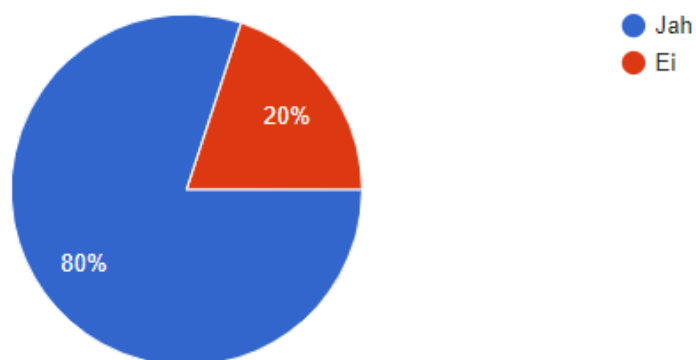
Joonis 8. Kui kaua olete Linuxit kasutanud?

Kas kasutate Linuxit regulaarselt? (Vähemalt 4 tundi päevas)



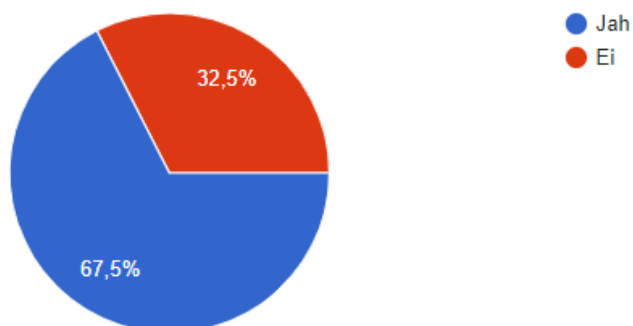
Joonis 9. Kas kasutate Linuxit regulaarselt? (Vähemalt 4 tundi päevas)

Kas Teil tekkisid probleemid pakettide paigaldamisel?



Joonis 10. Kas Teil tekkisid probleemid pakettide paigaldamisel?

Kas kasutaksite veebipõhist GUI-programmi, mis lihtsustab pakettide leidmist ja paigaldamist ühe käsuga?



Joonis 11. Kas olete veebipõhist GUI-programmi, mis lihtsustab pakettide leidmist ja paigaldamist ühe käsuga?

Lisa 3. Veebiteenuse väljundid

```
// http://46.101.206.218/api/map

[
  {
    "distro": "debian",
    "baseURL": "http://ftp.debian.org/debian/",
    "type": "debian-based",
    "releaseNames": [
      {
        "release": "bullseye",
        "components": [
          {
            "component": "main",
            "archs": [
              "all",
              "amd64",
              "arm64",
              "armel",
              "armhf",
              "i386",
              "mips64el",
              "mipsel",
              "ppc64el",
              "s390x"
            ]
          }
        ],
        {↔},
        {↔}
      ]
    },
    {
      "release": "buster",
      "version": "10.7",

```

Joonis 12 Koostatud "kaardi" väljund

```
// http://46.101.206.218/api/search?distro=debian&packages=%5B%22vim%22%5D&release=bullseye&arch=amd64
{
  "status": "success",
  "result": [
    {
      "Package": "vim",
      "Version": "2:8.2.1913-1+b2",
      "Installed_Size": "3231",
      "Maintainer": "Debian Vim Maintainers <team+vim@tracker.debian.org>",
      "Architecture": "amd64",
      "Depends": "vim-common (= 2:8.2.1913-1), vim-runtime (= 2:8.2.1913-1), libacl1 (>= 2.2.23), libc6 (>= 2.29), libcanberra0 (>= 0.2), libgpm2 (>= 1.20.7), lib3.1), libtinfo6 (>= 6)",
      "Description": "Vi IMproved - enhanced vi editor",
      "Homepage": "https://www.vim.org/ interface::text-mode, role::program, scope::application, uitoolkit::ncurses, use::editing, works-with::text, works-with::",
      "Section": "editors",
      "Filename": "pool/main/v/vim/vim_8.2.1913-1+b2_amd64.deb",
      "Size": "1463028",
      "Release": "bullseye"
    }
  ],
  "query": {
    "distro": "debian",
    "packages": "[\"vim\"]",
    "release": "bullseye",
    "arch": "amd64"
  }
}
}
```

Joonis 13 Paketi otsingu tulemus

```
// http://46.101.206.218/api/search?distro=debian&packages=%5B%22vim%22%5D&release=bullseye&arch=amd48
{
  "message": "Architecture amd48 not supported",
  "query": {
    "distro": "debian",
    "packages": "[\"vim\"]",
    "release": "bullseye",
    "arch": "amd48"
  },
  "status": "error"
}
}
```

Joonis 14 Paketi otsingu viga

```

#!/bin/sh -a

# Detect if user in Desktop Environment
HAS_DE="false"
if [ -z "$XDG_CURRENT_DESKTOP" ]; then
    HAS_DE="false"
else
    HAS_DE="true"
fi

# check and mark command for GUI notifications
HAS_ZENITY="false"
nullify command -v zenity && HAS_ZENITY="true"

# Function to display to user required messages
inform_user() {
    if [ $# -ne 2 ] && [ -z "$1" ] && [ "$2" != "info" ] && [ "$2" != "warning" ] && [ "$2" != "error" ]; then
        inform_user "Should be passed two arguments: message text and inform type(info, warning, error)" "error"
        exit 20
    fi

    if [ "${HAS_ZENITY}" = "true" ] && [ "${HAS_DE}" = "true" ]; then
        CMD="zenity --width 130 --title Scriptor --no-wrap --${2} --text '${1}'"
        eval ${CMD}
    else
        type=$(echo "$2" | awk '{ print toupper($0) }')
        printf "${type}: ${1}\n"
    fi
}

if [ "$(id -u)" != 0 ]; then
    inform_user "This script must be run as root" "error"
    exit 5
fi

```

Joonis 15 Paigalduskripti genereerimine

```
// http://46.101.206.218/api/generate?distro=debian&packages=%5B%22vim%22%5D&release=bullseye&arch=amd48
```

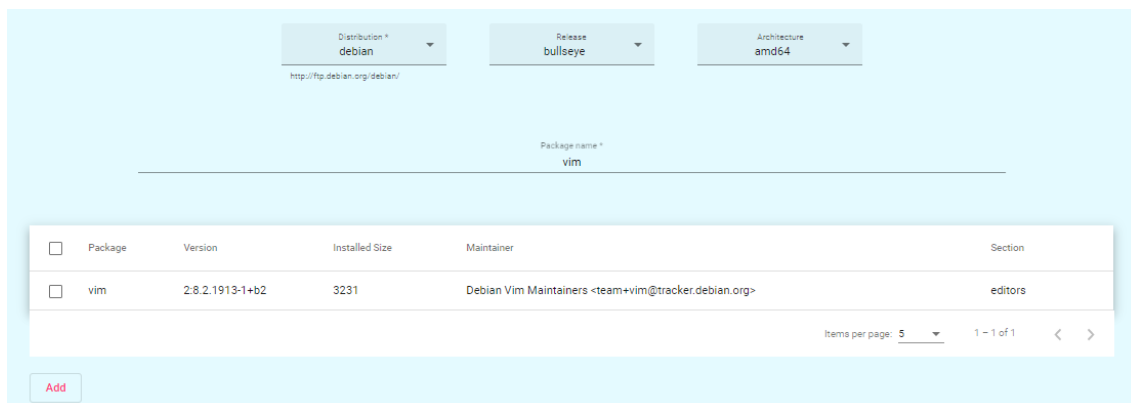
```

{
  "message": "Architecture amd48 not supported",
  "query": {
    "distro": "debian",
    "packages": "[\"vim\"]",
    "release": "bullseye",
    "arch": "amd48"
  },
  "status": "error"
}

```

Joonis 16 Paigalduskripti genereerimise viga

Lisa 4. Klientrakenduse vaatet

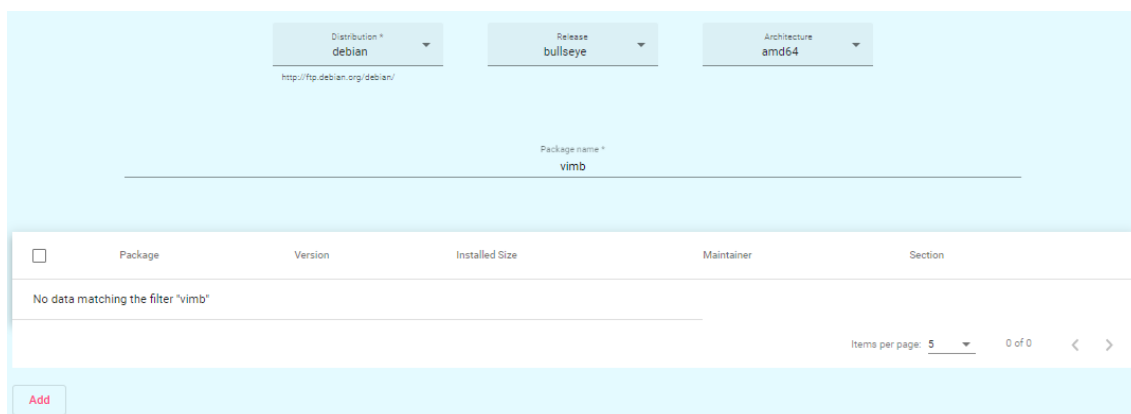


The screenshot shows a search interface with the following elements:

- Distribution: **debian** (with URL <http://ftp.debian.org/debian/>)
- Release: **bullseye**
- Architecture: **amd64**
- Package name filter: **vim**
- Table with columns: Package, Version, Installed Size, Maintainer, Section
- Table content:

Package	Version	Installed Size	Maintainer	Section
<input type="checkbox"/> vim	2:8.2.1913-1+b2	3231	Debian Vim Maintainers <team+vim@tracker.debian.org>	editors
- Footer: Items per page: 5, 1 - 1 of 1, navigation arrows
- Bottom left: **Add** button

Joonis 17 Paketi otsingu tulemus



The screenshot shows a search interface with the following elements:

- Distribution: **debian** (with URL <http://ftp.debian.org/debian/>)
- Release: **bullseye**
- Architecture: **amd64**
- Package name filter: **vimb**
- Table with columns: Package, Version, Installed Size, Maintainer, Section
- Table content:

Package	Version	Installed Size	Maintainer	Section
No data matching the filter "vimb"				
- Footer: Items per page: 5, 0 of 0, navigation arrows
- Bottom left: **Add** button

Joonis 18 Paketi otsingu sisutu tulemus

Distribution: debian

<input type="checkbox"/>	Package	Version	Installed Size	Maintainer	Section	Release	Architecture
<input checked="" type="checkbox"/>	vim	2.8.2.1913-1+b2	3231	Debian Vim Maintainers <team+vim@tracker.debian.org>	editors	bullseye	amd64
<input type="checkbox"/>	cowsay	3.03+dfsg2-8	92	James McDonald <james@jamesmcdonald.com>	games	bullseye	all
<input checked="" type="checkbox"/>	python3	3.9.1-1	191	Matthias Klose <doko@debian.org>	python	bullseye	amd64

Items per page: 5 1 - 3 of 3 < >

Execute following command:

```
curl -sg 'http://46.101.206.218/api/generate?distro=debian&packages=["vim","python3"]&release=bullseye&arch=amd64' | sudo -E sh
```

Joonis 19 Paigaldusskripti genereerimine

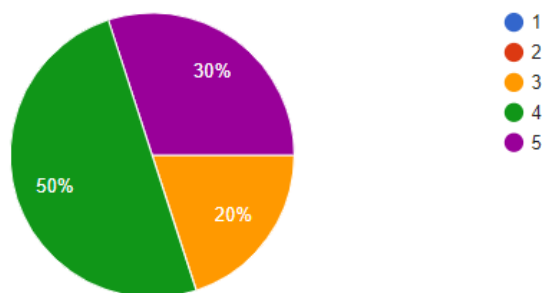
Generate Script

No package saved to generate script. Please use [search](#) to find and save packages.

Joonis 20 Ei valitud pakete paigaldusskripti genereerimiseks

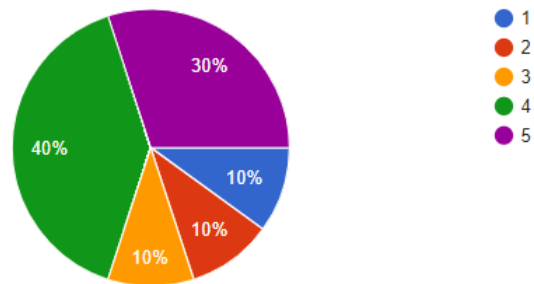
Lisa 5. Korduvküsimustiku graafikud

Kui rahul Teie olete pakettide leidmisega? (1- üldse ei ole rahul; 5- väga rahul)



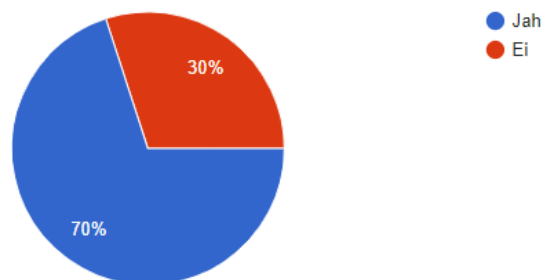
Joonis 21. Kui rahul Teie olete pakettide leidmisega?

Kui rahul olete pakettide paigaldamisega, kasutades allalaaditud skripti? (1- üldse ei ole rahul; 5- väga rahul)



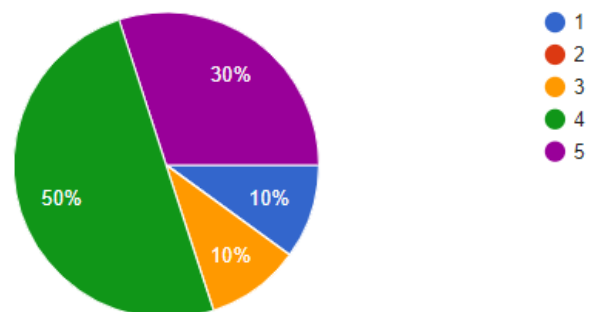
Joonis 22. Kui rahul olete pakettide paigaldamisega, kasutades allalaaditud skripti?

Kas pakutud teenus saab olla kasulik Linuxi kasutajate algajatele? (Näiteks et rohkem saada aru Linuxi tarkvarast, mängida skripti koodiga, mõista, kuidas töötab shell jne)



Joonis 23. Kas pakutud teenus saab olla kasulik Linuxi kasutajate algajatele?

Kas te soovitaksite seda teenust oma tuttavale/sõbrale? (1- ei ole nõus; 5- olen täiesti nõus)



Joonis 24. Kas te soovitaksite seda teenust oma tuttavale/sõbrale?