

TARTU ÜLIKOOL
LOODUS- JA TÄPPISTEADUSTE VALDKOND
Arvutiteaduse instituut
Informaatika õppekava

Magnar Vares
Valikkursuse „Linux ja Raspberry Pi” loomine
ja pilootimine
Bakalaureusetöö (9 EAP)

Juhendajad: Alo Peets, MSc
Eno Tõnisson, PhD

Tartu 2020

Valikkursuse „Linux ja Raspberry Pi” loomine ja pilootimine

Lühikokkuvõte:

Erinevate operatsioonisüsteemide käsitlemine on oluline oskus tehnoloogia valdkonnas. Selle bakalaureusetöö raames valmis valikkursus gümnaasiumiastmele, mis käsitleb Linuxi operatsioonisüsteemi distributsiooni Ubuntu näitel ja Raspberry Pi kasutamist. Kursuse maht on 32 õppetundi. Valminud valikkursus keskendub teoreetiliste teadmiste praktilisele rakendamisele. Bakalaureusetöös kirjeldatakse materjalide loomise protsessi, kursuse pilootimist Eesti gümnaasiumis ning pilootimise käigus kogutud eneseanalüüsi tulemusi.

Võtmesõnad: Linux, Raspberry Pi, operatsioonisüsteemid, õppematerjal, valikkursus

CERCS: P175 Informaatika, süsteemiteooria, S281 Arvuti õpiprogrammide kasutamise meetodika ja pedagoogika

Creation of Upper Secondary School Elective Course “Linux and Raspberry Pi”

Abstract:

Understanding different operating systems is important in informatics. The aim of this Bachelor's thesis was to conduct a study material covering the Linux operating system and Raspberry Pi. The main focus in materials is on Linux distribution Ubuntu. The materials are highly practical so that students can get the hands-on feeling. The course consists of 32 lessons, each lesson is 45 minutes. The need of course materials in the Estonian language covering Linux, the conducting process of the course materials and conducted materials are described in the first part of the thesis. The second part of the thesis focuses on piloting the course in Estonian upper secondary school and the results of the teacher's self-analyses.

Keywords: Linux, Raspberry Pi, operating systems, study material, elective course

CERCS: P175 Informatics, systems theory, S281 Computer-assisted education

Sisukord

Sissejuhatus	5
1. Materjalide vajalikkus	7
1.1 Projekt “Noored Koodi”	7
1.2 Linuxi tundmise tähtsus IKT valdkonnas	8
2. Tarkvara ning riistvara valikud	9
2.1 Operatsioonisüsteem Linux	9
2.2 Distributsioon Debian	10
2.3 Distributsioon Ubuntu	10
2.4 Virtualiseerimistarkvara VirtualBox	11
2.5 Miniarvuti Raspberry Pi	12
2.6 Raspberry Pi lisatarvikud	12
2.6.1 Puuteekraan	13
2.6.2 Laienduskomplekt CamJam EduKit 2 – Sensors	13
2.6.3 Laiendusplaat SenseHat	13
2.6.4 Kaamera moodul	13
2.7 Programmeerimiskeel Python	14
2.8 Ekraanipildi lindistamise programm Peek	14
3. Materjalide loomine	15
3.1 Materjalide loojad	15
3.2 Teemavalikud kursuse Linuxi osale	15
3.2.1 Linuxi materjalide loomise protsess	17
3.2.2 Linuxi materjalide näidistund “Käsurida 202”	17
3.3 Raspberry Pi materjalid	21
4. Kursuse materjalide pilootimine	23
4.1 Tunni korraldus ning õpetaja roll tunnis	23

4.1.1 Linux'i kontrolltöö	24
4.1.2 Raspberry Pi projekti teostamise plaan	24
4.2 Pilootimise tulemuste hindamine	25
4.2.1 Eneseanalüüsi vormi loomine	25
4.2.2 Pilootimise tulemused eneseanalüüsi ning vaadeldud õpetaja tagasiside põhjal	26
5. Kokkuvõte	28
Viidatud kirjandus	29
Lisad	31
I. Kursuse materjalide loomise tööjaotus	31
II. Valminud kursuse materjalid	33
III. Tunni KÄSURIDA 202 materjal	34
IV. Eneseanalüüsi küsimustik tunni "Linux'i töölaud" näitel	48
V. Litsents	51

Sissejuhatus

Infotehnoloogia valdkonna populaarsus on 21. sajandil olnud pidevas kasvus. Ka Tartu Ülikooli informaatika õppekavale vastuvõetute arv on liikunud kasvutrendis, 2018. aastal vastuvõetute arv on võrreldes 2011. aasta vastuvõetutega kaks korda suurem [1]. Ühtlasi tähtsustatakse aina enam infotehnoloogia teadmiste õpetamist nii põhi- kui ka keskkoolis, Haridus- ja Teadusministeerium on käivitanud seda toetava programmi “Digipööre” [2]. Viimase paari aastaga on valminud või veel valmimisjärgus mitmed keskkoolidele suunatud valikkursused. Hariduse Infotehnoloogia Sihtasutuse eestvedamisel on välja töötatud gümnaasiumi informaatika ainekava, mille eesmärk on arendada õpilaste IT-oskusi ning lahenduste leidmist elulistele probleemidele [3]. Ka koolid panustavad, võimaldades aina enam valminud materjalide järgi õppimist. Vaadates erinevaid valminud materjale, tuleb selgelt välja, et hetkel on puudus Linuxi operatsioonisüsteeme käsitletavatest kursustest. Autorile teadaolevalt on värskeim mahukas eestikeelne veebipõhine õppematerjal Linuxi kohta Kaido Kikkase kirjutatud “Pingviiniaabits”, mis valmis 2004. aastal. Kuna see õpik on pikk ja sisutihe, tuleks selle põhjal kursuse andmiseks põhjalikult valida sobivad peatükid. Ühtlasi on kuueteistkümne aastaga muutunud palju nii Linuxi kui ka üldisemalt tehnoloogia maastikul ja autori hinnangul on “Pingviiniaabits” iganenud. Internetist võib leida mitmeid eestikeelseid artikleid, mis räägivad Linuxist, aga autorile teadaolevalt ei leidu eestikeelseid üldhariduskooli sobivaid põhjalikke veebikursusi või õppematerjale, mis ei ole aegunud. Seega peaksid tahes-tahtmata kõik Linuxi-huvilised kas usaldama vananenud materjale või otsima võõrkeelseid materjale.

Sarnane probleem on ka Raspberry Pidega. Mitmed Eesti koolid omavad mõnda või ühe klassi jagu Raspberry Pi miniarvuteid. Autori kogemusel ei kasuta osa nendest koolidest Raspberry Pisid, kuna taaskord on puudus materjalidest. Kui koolil ei ole õpetajat, kellel oleks vajalikke teadmisi või aega teemakohaste materjalide loomiseks, on selge, et vahendid jäävad seisma. Bakalaureusetöö raames valmiv kursus sisaldab ka Raspberry Pi materjale kahel põhjusel – esiteks on Raspberry Pi hea miniarvuti, millel kasutada Linuxi operatsioonisüsteemi, ning teiseks võimaldaksid valmivad materjalid koolidel mugavamalt kasutada Raspberry Pisid õppetöös. Bakalaureusetöös ei käsitleta Linuxi ja Raspberry Pi materjale võrdses mahus, kuna loomise käigus oli põhirõhk kursuse suuremal osal ehk Linuxi teemal.

Ühtlasi kasutatakse Raspberry Pi tundides hästi koostatud ning illustreeritud olemasolevaid juhendeid, mis koondatakse ühtseks materjaliks.

Bakalaureusetöö raames valmis valikkursus, mis õpetab Linuxi operatsioonisüsteemi käsitlemist ning arendab õppija silmaringi ka teiste operatsioonisüsteemide eripärade osas. Valikkursuse teine osa õpetab Raspberry Pi arvuti kasutamist. Eesmärk oli luua eelteadmisteta läbitav ning praktiline materjalide kogu. Materjalide loomisel arvestati õpetamise mugavust nii tehnoloogiliste vajaduste kui ka õpetaja vajalike teadmiste osas, et suurendada materjalide kasutamise võimalusi erinevates Eesti koolides. Õppematerjalid loodi selle bakalaureusetöö autori ning Marten Tamme koostöös, tööjaotus on välja toodud tööjaotustabelis (lisa I). Kursust pilooditi ühes Eesti gümnaasiumis, et selle põhjal hinnata materjalide kasutatavust.

See töö koosneb viiest osast. Esimeses osas kirjeldatakse valminud töö raames kasutatud riist- ning tarkvara. Teises osas tutvustatakse projektist “Noored Koodi” ning põhjendatakse “Linuxi ja Raspberry Pi” kursuse vajadust. Kolmandas osas kirjeldatakse teemade valimise meetodikat ning materjalide loomise protsessi. Neljandas osas antakse ülevaade kursuse pilootimisest, tagasisidest kursusele õpetaja eneseanalüüsi vormis ning tehakse kokkuvõtte pilootimise tulemustest. Viiendas osas tehakse kokkuvõtte valminud kursuse loomisest ning tuuakse välja selle võimalikud edasiarendused ning kasutuskohad.

1. Materjalide vajalikkus

Töö sissejuhatuses toodi välja, et hetkel on puudus terviklikest eestikeelsetest Linuxi operatsioonisüsteemi puudutavatest materjalidest. Selles peatükis antakse ülevaade projektist “Noored Koodi”, mille raames see valikkursus valmis. Seejärel tuuakse välja peamised argumendid, miks on Linuxit käsitlevat kursust vaja.

1.1 Projekt “Noored Koodi”

“Noored Koodi” on Tartu ülikooli arvutiteaduste instituudi projekt, mida rahastati meetme TEEME+ raames. TEEME+ on sihtasutus Archimedese koordineeritav meede, millega toetatakse loodus- ja täppisteaduste ning tehnoloogia valdkonda populariseerivaid tegevusi noortele [15]. “Noored Koodi” projekti eesmärk on laiendada kooliõpilaste teadmisi info- ja kommunikatsioonitehnoloogia (edaspidi IKT) valdkondades. Seejuures valmis seitse töötuba, mida viidi läbi Eesti koolides rohkem kui 250. korral. Ühtlasi korraldatakse 15 huvilaagrit ning kolm üle-eestilist võistlust. Töötubadele ning laagritele lisaks on valminud kokku kuus gümnaasiumiastmele suunatud valikkursust:

1. Virtuaalreaalsuse õppekava 45 min x 16;
2. Turvalisuse õppekava 45 min x 16;
3. Algoritmika õppekava 45 min x 16;
4. Veebilehtede õppekava 45 min x 16;
5. Linuxi (Raspberry Pi) õppekava 45 min x 32;
6. Arduino õppekava 45 min x 32.

Bakalaureusetöö autor on selles projektis kaasa löönud alates 2017. aasta detsembrist ning selle aja jooksul läbi viinud 75 töötuba ja olnud kahel võistlusel kohapealseks koordinaatoriks ning läbiviijaks. Bakalaureusetöö autorile pakkus projektijuht Heilo Altin võimalust koostöös Marten Tammega luua Linuxi ja Raspberry Pi kursuse materjalid. Õppematerjalid valmisid paaristööna, kus lõplik vastutus ja otsustusõigus oli bakalaureusetöö autoril.

1.2 Linuxi tundmise tähtsus IKT valdkonnas

Linuxi operatsioonisüsteemil töötab NetMarketShare'i andmetel 1.89% arvutitest, seejuures Windows omab 88.14% ning Mac OS 9.38% turuosast [16]. Nende andmete põhjal võib jääda mulje, et Linux on väga väikest osakaalu omav operatsioonisüsteem. See valikkursus on suunatud õpilastele, kes tunnevad huvi IKT valdkonna vastu. Seega võib ka eeldada, et kursuse valivad oma õppekavasse õpilased, kelle õppekavas leidub ka teisi informaatika-alaseid aineid. Ühtlasi on suurem tõenäosus, et need õpilased tegelevad iseseisvalt programmeerimisega ning valivad kõrg- või kutsekoolis IKT eriala. Nii Tartu Ülikooli informaatika eriala kui ka Tallinna Tehnikaülikooli informaatika eriala õppekavasse kuulub aine "Operatsioonisüsteemid" [17, 18]. Mõlemas kõrgkoolis õpitakse operatsioonisüsteemide aines tundma Linuxit, kuid mitte operatsioonisüsteemi igapäevast kasutamist, vaid eriala spetsiifilisi oskusi. Seega on aine läbimisel eelis tudengitel, kellel on olemas varasem kokkupuude Linuxiga.

Linux on populaarne valik veebilehtede serveri operatsioonisüsteemiks, omades W3Techsi uuringu andmetel 60% turuosast [19]. StackOverflow koostatud arendajatele suunatud uuringust võttis 2019. aastal osa 90 tuhat inimest ning nendest 25.6% vastasid, et nemad kasutavad tööarvutis Linuxi operatsioonisüsteemi, 26.8% vastanutest kasutab Mac OSi ning 47.5% Windowsi operatsioonisüsteemi. Viiskümmend protsenti vastanutest ütles, et nad on möödunud aasta jooksul teinud arendustöid Linuxi platvormile. Ühtlasi on Linuxit kasutavad arendajad enim rahul oma operatsioonisüsteemiga (kui Linuxi kasutajatest on rahul oma operatsioonisüsteemiga 83.1%, siis Windowsil on see protsent 64.2%) [20]. Noori tööturuks ettevalmistamisel on oluline pakkuda võimalikult laia valikut neid huvitavast valdkonnast. Noorte ettevalmistamisel tööturuks on oluline pakkuda laia valikut teadmisi neid huvitavast valdkonnast. Eelneva põhjal on näha, et Linuxi operatsioonisüsteem on tarkvara arenduses olulisel kohal. Seetõttu on oluline pakkuda noortele teemakohaseid materjale.

2. Tarkvara ning riistvara valikud

Selles peatükis kirjeldatakse bakalaureusetöoga seotud tarkvara ning riistvara, mida kasutati õppematerjali loomise käigus või mida kasutatakse õppematerjalis. Tarkvara valikul eelistati üldiselt tunnustatud vabavaralisi lahendusi.

2.1 Operatsioonisüsteem Linux

Linux on vabavaraline operatsioonisüsteemi tuum, mille loojaks on soomlane Linus Torvalds. L. Torvalds alustas hobiprojektina Linuxi tuuma loomist 1991. aastal ja samal aastal valminud Linuxi tuuma versiooninumbriga 0.02 peetakse esimeseks ametlikuks väljalaskeks. Linus Torvaldsi poolt 1992. aastal koostatud artikli põhjal mõjutasid varasemad operatsioonisüsteemid UNIX ja selle põhjal loodud Minix tugevalt Linuxi väljatöötamist [4].

Kuigi nimi “Linux” viitab tuumale, nimetatakse tänapäeval Linuxi tuuma kasutatavaid operatsioonisüsteeme lihtsalt Linuxiteks. Selline nimetus on tekitanud vaidlusi. Free Software Foundation on liikumine, mille eesmärk on pakkuda tasuta, vabavaralist operatsioonisüsteemi ning tagada inimeste vabadus muuta ning arendada seda süsteemi. Nende arendatava operatsioonisüsteemi nimi on GNU. Linuxiks nimetatavad operatsioonisüsteemid kasutavad üldjuhul GNU erinevaid komponente ning seetõttu tuleks Free Software Foundation’i arvates nimetada selliseid operatsioonisüsteeme nimetusega “GNU/Linux operatsioonisüsteem” [5]. Sellele vaatamata kasutatakse kirjanduses enamasti üldnimetusena nime “Linux”. Selles bakalaureusetöös ning bakalaureusetöö raames valminud õppematerjalides mõistetakse nime “Linux” all toimivat operatsioonisüsteemi, mis kasutab Linuxi tuuma.

Linuxi distributsioon (kõnekeeles “distro”) on üks konkreetne versioon GNU/Linux operatsioonisüsteemist. Distributsioonist võib mõelda kui eraldiseisvast operatsioonisüsteemist (nagu Windows), millel on üldjuhtudel olemas vabalt saadav süsteemitarkvara paigaldusfail, paketi haldur ning töölaua keskkond [6]. Lehekülje DistroWatch andmetel eksisteerib hetkel üle 270 erineva distributsiooni [7]. Kursuse materjalides käsitletakse põhjalikult neist vaid Ubuntu distributsiooni. Kursuse teises pooles kasutatakse Raspberry Pi-de jaoks loodud distributsiooni Raspian.

2.2 Distributsioon Debian

Järgnev lõik tugineb allikal [8]. Debiani-nimelise distributsiooni arendus algas 1993. aastal Ian Murdocki eestvedamisel. Tegu on Linuxi kernelil põhineva operatsioonisüsteemiga. Selle operatsioonisüsteemi arengusse on panustanud väga paljud arendajad üle maailma, kes teevad seda oma isiklikust huvist. Kursusel kasutatav distributsioon Ubuntu on välja kasvanud Debianist. Ubuntu ja Debiani arendus on endiselt tugevalt seotud ning mitmed Ubuntu arendajad panustavad jätkuvalt ka Debiani distributsiooni arendamisse.

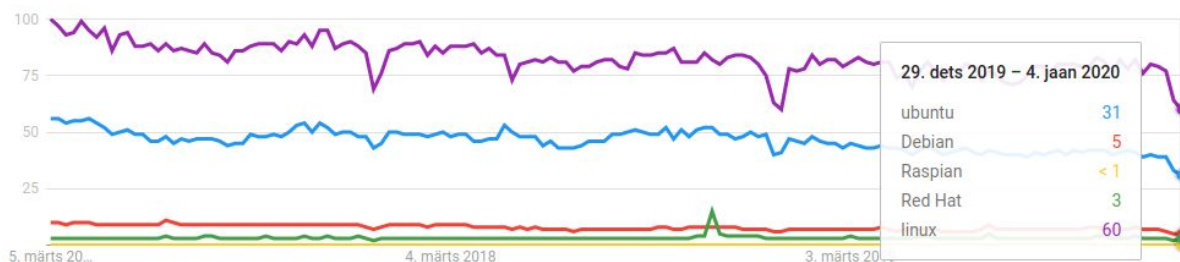
2.3 Distributsioon Ubuntu

Järgnev lõik tugineb allikal [9]. Sõna “ubuntu” tähendab suulu ja koosa keeles “ma olen see, kes ma olen selle tõttu, millised meie oleme” (inglise keeles “I am what I am because of who we all are”). Ubuntu distributsiooni loojad on selle tähenduse üle kandnud oma distributsiooni, kuhu üritatakse koondada parim tarkvara, mida programmeerijate kogukond maailmaga jagab. Linuxi distributsioon Ubuntu sai alguse 2004. aastal, kui grupp Debiani-nimelise distributsiooni arendajaid asutasid ettevõtte Canonical. Sama aasta oktoobris anti välja esimene Ubuntu versioon ning Ubuntu sai esimene distributsioon, millel ilmub uus versioon iga kuue kuu tagant. Iga neljas ehk iga 24 kuu tagant väljastatav Ubuntu versioon sisaldab oma nimes lühendit LTS (Long Term Support), mis tähendab, et nendele versioonidele toimub pikaajaline tugi ning nende versioonide täiendamisega tegeletakse vähemalt viis aastat alates esimesest väljalaskest.

Kursusel kasutatakse Ubuntu versiooni 18.04 kuna tegu oli viimase Ubuntu LTS väljalaskega materjalide loomise ajal (suvi 2019). Bakalaureusetöö kirjutamise ajal ilmus ka Ubuntu uus LTS versioon 20.04. Ubuntu distributsiooni valimise kasuks rääkis mõlema autori isiklik huvi selle distributsiooni vastu. Mõlemad õppematerjalide loojad kasutasid materjalide valmimise ajal isiklikus arvutis operatsioonisüsteemi Ubuntu.

Peamine põhjus Ubuntu käsitlemiseks materjalides on selle populaarsus. Distributsioonide populaarsust on raske hinnata. Steven J. Vaughan-Nicholsi sõnul saavad operatsioonisüsteemide statistikat veebilehe külastamise kaudu hindavad organisatsioonid mugavalt teada, kas külastaja operatsioonisüsteem on Linux, kuid täpset distributsiooni välja ei selgitata [10].

Google'i statistikat¹ kasutades on lihtne näha, et otsingud, mis sisaldavad võtmesõna “Ubuntu”, on kordades populaarsemad kui otsingud, mis on seotud teiste enamlevinud Linuxi distributsioonidega (joonis 1). Üle 35% veebilehekülgedest kasutavad serveri operatsioonisüsteemina Ubuntu [11].



Joonis 1. Populaarsete Linuxi distributsioonide otsingute statistika Google Statistics andmetel¹.

On selgelt näha ning ka tehnikaartiklites läbivalt tunnistatud², et Ubuntu on levinuim distributsioon algajatele Linuxi kasutajatele. Ühtlasi on Ubuntu peamine distributsioon, mida soovitatakse algajale kasutajale. Ubuntu kogukond on väga suur, mille parimaks näiteks on internetilehekülg Ask Ubuntu³, mille eesmärk on ühendada Ubuntu-ga seotud murede või küsimustega arvutikasutajad teiste huvilistega, kes oskavad abi pakkuda. Selle lehekülje on loonud ettevõtte Stack Overflow⁴, mis on loonud pea sadakond erinevat valdkonda või teemat puudutavat sarnast küsimuste-vastuste lehekülge. Ubuntu on ainuke Linuxi distributsioon, mille kohta on selline temalehekülg loodud. Mitmed teised üldlevinud distributsioonid on sarnased Ubuntu-le või põhinevad Ubuntu distributsioonil (näiteks Linux Mint⁵).

2.4 Virtualiseerimistarkvara VirtualBox

Kursuse loomisel tuli arvestada asjaoluga, et valikkursust õpetatakse kasutades kooli arvuteid. Eestis kasutavad kooliarvutid üldjuhul Windows'i operatsioonisüsteemi. Ei saa eeldada, et koolid vahetaksid ühe valikkursuse õpetamiseks arvutite operatsioonisüsteemi Ubuntu vastu. Ühtlasi tehakse kursuse raames läbi rohkelt praktilisi harjutusi, mille valesti või vales järjekorras täitmine võib operatsioonisüsteemi rikkuda. Seega kasutatakse kursusel

¹<https://trends.google.com/trends/explore?date=2017-03-01%202020-01-03&q=ubuntu,Debian,Raspian,Red%20Hat,linux>

²<https://www.howtogeek.com/191207/10-of-the-most-popular-linux-distributions-compared/>, <https://itsfoss.com/best-linux-distributions/>, <https://www.fossmint.com/best-linux-desktop-distros-of-2018/>

³<https://askubuntu.com/>

⁴<https://stackoverflow.com/>

⁵<https://linuxmint.com/>

operatsioonisüsteemi virtualiseerimise tarkvara VirtualBox⁶, mis paigaldatakse kooli arvutis olevasse operatsioonisüsteemi. VirtualBox on tasuta, vabavaraline ning GNU litsentsi all väljastatav tarkvara. Selle tarkvara abil luuakse virtuaalmasin, kuhu paigaldatakse Ubuntu distributsioon. VirtualBox on paindlik ning kiire, võimaldades luua ning ka samaaegselt käivitada mitut erinevat (või samal) operatsioonisüsteemil töötavat virtuaalmasinat. Seetõttu on Virtualboxi abil virtualiseeritud Ubuntu operatsioonisüsteemil töötav arvuti hea variant Linuxi õppimiseks arvutis, mis kasutab mõnda teist operatsioonisüsteemi. Ühtlasi töötab VirtualBox probleemideta nii Windowsi kui ka MacOSi arvutites, mis võimaldab soovi korral õpilastel kasutada kursuse raames oma isiklikku sülearvutit.

2.5 Miniarvuti Raspberry Pi

Kursuse teises osas kasutatakse pangakaardi suurust miniarvutit Raspberry Pi, mille esimene versioon valmis 2012. aastal. Raspberry Pi väljatöötamise eesmärgiks oli luua odav, kergesti kättesaadav ning kasutatav miniarvuti, mille abil saavad õpilased ning teised soovijad õppida programmeerimist ning arendada insenerioskusi [12]. Kursuse loomisel kasutati mudelit Raspberry Pi 3 Model B, kuid materjal on väikeste kohandustega läbitav ka teiste mudelitega alates Raspberry Pi teisest versioonist. Näiteks Raspberry Pi teise versiooni mudelitel puudub sisseehitatud WiFi adapter, seega tuleb internetiga ühendamiseks kasutada kaablit või juurde muretseda USB WiFi adapter.

2.6 Raspberry Pi lisatarvikud

Kursuse raames kasutatakse mitmeid lisatarvikuid, et laiendada õpilaste silmaringi ning võimaldada huvitavamate projektide tegemist. Kasutatavate lisatarvikute nimekiri koostati koostöös juhendaja Alo Peetsiga ning arvesse võeti erinevate projektide raames Eesti koolide tellitud Raspberry Pi lisatarvikute komplekte, et vähendada valikkursuse õpetamisel võimalikku probleemi riistvara puudumise osas. Ühtlasi on peaaegu kõigi lisatarvikute puudumise korral välja pakutud alternatiivne lahendus materjali läbimiseks.

⁶ <https://www.virtualbox.org/>

2.6.1 Puutekraan

Raspberry Pi puutetundlik ekraan on seitsmetolline miniarvuti tootjate loodud ekraan, mis on võimalik kinnitada miniarvuti külge kruvide ning vajalike juhtmete abil. Puutekraani on võimalik kasutada mitmesugustes isiklikes projektides. Näiteks luues köögiseinale süsteemi, kus puhkerežiimis jooksevad perepildid ning soovi korral saab lugeda toitude retsepte. Õpilase otsene kogemus puutekraaniga aitab laiemalt mõelda selle kasutusladele. Ühtlasi on puutekraani kasutamine mugavam võrreldes kooliarvuti ekraani kasutamisega, sest see võimaldab kasutada klassiarvutit tööjuhendi vaatamiseks ning kaugühenduse loomiseks 15. praktikumis.

2.6.2 Laienduskomplekt CamJam EduKit 2 – Sensors

CamJam EduKit 2 on komplekt, milles sisalduvad kaks LED-lampi, liikumisandur, maketeerimislaud, takistid ning piisav kogus juhtmeid kõige eelneva ühendamiseks. Selles komplektis olevad LED-tuled leiavad kasutust pea kõigis Raspberry Pi tundides ning liikumisandurit kasutatakse turvakaamera teostamiseks. CamJam EduKit 2 on loodud üritustesarja Cambridge Raspberry Jam korraldajate ja ettevõtte The Pi Hut koostöös [13].

2.6.3 Laiendusplaat SenseHat

SenseHat on Raspberry Pi viikudele kinnituv moodul, millel asuvad mitmed sensorid, 8x8 RGB LED-tulede maatriks ning juhtkang. LED-tuledest koosnevat maatriksit on kerge programmeerida kuvama erinevaid tekste ning võimaldab ka keerukamate ülesannete püstitamist.

2.6.4 Kaamera moodul

Kursuse raames kasutatakse ka Raspberry Pi loojate valmistatud kaameramoodulit, mis ühendub kergesti miniarvuti külge. Selle kaameraga pildistamine on programmeerimiskeele Python abil lihtne ning loogiline. Seda moodulit kasutatakse näiteks turvakaamera teostamiseks, aga ka muudes ülesannetes-projektides.

2.7 Programmeerimiskeel Python

Python on programmeerimiskeel, mis on populaarne valik kõrg-, kutse- ja keskkoolides programmeerimise algõppe tundides. Keele süntaks on lihtne ning loogiline, mistõttu on see programmeerimiskeel algajatele lihtsalt õpitav. Raspberry Pi tundides tuleb igas tunnis kasutada Pythonit programmeerimiseks. Seejuures on arvestatud, et õpilane ei pruugi olla varem programmeerimisega kokku puutunud, kuid osavamatele on olemas laialdaselt lisäülesandeid.

2.8 Ekraanipildi lindistamise programm Peek

Peek on GNU litsentsi all välja antud vabavaraline ning avatud lähtekoodiga programm, mis on mõeldud kasutaja määratud ekraaniala lindistamiseks. Programmi loomise eesmärk oli looja sõnul võimaldada mugavalt näiteks oma loodud programmi demonstreerimiseks või mõne programmi veasituatsiooni lindistamiseks, et seeläbi aidata programmi arendajatel mõista, kus viga esineb [14]. Seda programmi kasutati ekraanilindistuste tegemiseks materjalide loomise käigus. See lahendus võimaldas visuaalsete näidete toomisel kasutada piltide asemel GIF-vormingus lindistusi, mis annavad täpsemalt edasi materjalide sisu.

3. Materjalide loomine

Selles peatükis keskendutakse materjalide (lisa II) valmimise protsessile. Esmalt avatakse materjalide loojate tausta, millel on autori arvates oluline roll kursuse sisu juures. Seejärel kirjeldatakse tehtud teemavalikuid ning materjalide loomise protsessi kahes osas – esmalt kirjeldatakse Linuxi teemasid ning seejärel Raspberry Pi teemasid.

3.1 Materjalide loojad

Materjalide autoriteks on Marten Tamm (Tartu Ülikooli informaatika bakalaureuseõppe teine aasta) ja Magnar Vares (Tartu Ülikooli informaatika bakalaureuseõppe kolmas aasta). Valikkursuse materjalide loomine võeti ette kahel põhjusel. Esiteks soovisid mõlemad autorid omandada kogemust tervikliku õppematerjali koostamises. Teiseks oli mõlemal autoril huvi Linuxi operatsioonisüsteemi vastu ning soov selle kohta rohkem õppida. Ühtlasi loodeti seeläbi luua võimalikult kaasahaaravad ning õppijast lähtuvad materjalid. Kaks kuud enne materjalide kirjutamise juurde asumist valisid mõlemad autorid oma arvuti operatsioonisüsteemiks Linuxi distributsiooni Ubuntu versiooninumbriga 19.04. Lisaks igapäevase arvutikasutuse käigus õpitule kogusid mõlemad autorid Linuxi kohta võimalikult palju teadmisi enne materjalide loomisega alustamist. Materjalid valmisid 2019. aasta suvel. Materjalide kirjutamise ajaks olid autorid Linuxiga hästi kursis. Muuhulgas oli lõputöö autor läbinud informaatika õppekavasse kuuluvad ained “Operatsioonisüsteemid” ning “Andmeturve”, mis käsitlevad samuti Linuxi operatsioonisüsteemi. Kuna suurem osa Linuxi teadmistest oli autoritel kogunenud läbi oma kogemuste õppides, võtsid autorid kursuse loomisel eesmärgiks käsitleda suures mahus igapäevase kasutusega seotud teemasid. Sellise eesmärgi põhjal loodud materjalid on autorite arvates õppijatele esiteks huvitavad, kuna seostuvad õppija senise operatsioonisüsteemide kasutamise kogemusega ning teiseks väärtuslikumad, kuna õpitud teadmised tulevad Linuxi kasutamisel suurema tõenäosusega kasuks.

3.2 Teemavalikud kursuse Linuxi osale

Kursuse Linuxi osa hõlmab endast kümmet 90-minutilist tundi, kus iga tund on jaotatud kaheks 45-minutiliseks ploki. Esialgsed ideed ning teemad, millest lähtuda, pakkus välja

bakalaureusetöö juhendaja Alo Peets. Juhendaja pakutud teemade nimekiri ei täpsustanud praktikumide täpsemat sisu ning kattis alla poole mahust. Materjalide autorid tegid lõpliku teemavaliku tuginedes oma kogemusele Linuxi õppimisel. Alo Peets kohtus kursuse autoritega materjalide loomise ajal regulaarselt ning andis tagasisidet materjalide teemavalikule. Tunni teemade valikul peeti oluliseks, et kaetud saaksid Linuxi tundmise kõige olulisemad alustalad ja põhjused, miks eelistatakse Linuxit teistele operatsioonisüsteemidele. Kursuse esimesed kümme teemat koos tunni sisu kirjeldava lausega:

1. Sissejuhatus Linuxisse. VirtualBoxi installeerimine, Ubuntu distributsiooni installeerimine virtuaalmasinasse ning esimene käivitamine.
2. Linuxi töölaud. Graafilise liidese kasutamine ja kujundamine.
3. Käsuri 101. Lihtsamate käsurea käskude õppimine.
4. Ubuntu tarkvara. Erinevate paigaldusviiside õppimine programmide installeerimiseks ning võrdlus Windowsi programmidega.
5. Käsuri 202. Aliaste ning skriptide loomine, nende kasutamine ning käsurea parem tundmine.
6. Failiõigused ja käsurea kordamine. Linuxi failipuu turvalisuse kujundamine ning erinevate kasutajate õigused failipuus.
7. Turvalisus. Superkasutaja loomine, Linuxi masina parooli murdmine ning arvuti füüsilise turvalisusega tutvumine.
8. Windowsi programmid Linuxis ja töölaua keskkonnad. Windowsi erinevate programmide tööle panek Ubuntu ning erinevate töölaualeskkondade aktiveerimine ning kasutamine.
9. Kordamine ja kuidas Linuxit oma arvutis tööle panna. Võimalus korrata eelnevaid teemasid ning uurida lähemalt, kuidas koduses arvutis operatsioonisüsteemi vahetada saab.
10. Linuxi kontrolltöö. Selles tunnis teevad õpilased kontrolltööd läbitud teemade peale.

Teemavalik otsustati ära enne materjalide loomise protsessi. Ainus materjalide loomise ajal valitud teema oli “Windowsi programmid Linuxis ja töölaua keskkonnad”. Teemade järjekord valiti põhimõttel, et uusi omandatud teadmisi oleks võimalus tulevastel tundidel uuesti ära kasutada. See tagab teadmiste kordamise ning seeläbi teadmiste parema kinnistumise. Ei saa

eeldada, et iga õpilane on tunnis alati kohal või kõik õpilased lahendavad praktilisi ülesandeid sama kiirusega. Seega on enne kontrolltööd toimivas kordamistunnis õpilasel võimalus üle korrata või järele teha läbimata teemasid. Hästikoostatud kontrolltöö, mis sisaldab praktilisi ülesandeid, mõjub samuti hea teadmistekinnistajana ning võtab kokku läbitud teemad.

3.2.1 Linuxi materjalide loomise protsess

Materjalide autorid tegid iga teema väljatöötamisel pidevalt koostööd. Esialgse peatüki kirjapanek oli teemade kaupa jaotatud järgnevalt:

- Marten Tamm - kolmas, neljas, kaheksas ja üheksas peatükk,
- Magnar Vares - esimene, teine, viies, kuues ja seitsmes peatükk.

Esmalt kasutati materjalide loomiseks ühist Google Drive kausta, mis võimaldas ühe peatüki samaaegset uuendamist. Töökäigu jälgimiseks kasutati tööjaotustabelit, kus uuendati regulaarselt iga peatüki valmimise järku. Kui peatüki materjal sai esialgsel vastutajal valmis, siis vaatas selle üle kaasautor, kes andis tagasisidet ülesannetele ja sisule. Seejärel toimus peatüki uuendamine vastavalt saadud tagasisidele. Kui peatükk oli tagasiside põhjal uuendatud, loeti see valminuks. Selline protsess toimus peatükkide lõikes paralleelselt ning samal ajal olid valmimas mitu erinevat peatükki. Kui kõik Linuxi teemat katvad kümme peatükki olid valminud (v.a kontrolltöö), luges Magnar Vares peatükid uuesti läbi. Selle läbilugemise käigus keskenduti faktide kontrollimisele ning materjalide loogilisele järjestusele. Vajadusel muudeti töö käigus tundide sisu väiksemal määral. Näiteks muudeti peatükis “Käsurida 101” õpitavate käskluste järjestust loogilisemaks (kõigepealt tuleb osata luua faile, seejärel õppida neid kustutama). Peale põhjalikku ülelugemist liigutati materjalid Google Drive’ist Courses-keskkonda.

3.2.2 Linuxi materjalide näidistund “Käsurida 202”

Kõigi tundide ülesehitamisel peeti peamiselt silmas praktilist tegevust. Teoorialõikude vahele on pandud praktilisi näiteid või ülesandeid, mida saavad õpilased realselt järele katsetada. Ülesannete mõtlemisel katsusid materjalide autorid läheneda eluliselt ning luua seoseid õpilaste varasema arvutikasutamisega. Järgnevalt tuuakse näide Linuxi teemade tunni ülesehitusest viienda osa ehk “Käsurida 202” (lisa III) näitel.

Osa “Käsurida 202” eesmärk on kinnistada õpilaste seniseid käsurea oskusi ning teha esmane tutvus aliaste ja skriptidega. Sarnaselt teistele peatükkidele on peatükk jaotatud kaheks 45-minutiliseks tunniks. Eesti gümnaasiumites on tundide pikkused erinevad, rakendatakse 75-minutilise, 45-minutilise ning 90-minutilise tunde (vahel ka kombinatsioone nendest). Seega tundus kõige paremini sobivat variant, kus õpetaja saab kasutada keskel läbi 45-minutilise tunde vastavalt kooli tunniplaanile.

Peatüki “Käsurida 202” esimene tund algab sissejuhatava ning illustreeriva jutuga, kus anonüümne isik internetist räägib, kuidas ta õppis mõistma, miks pikaajalised Linuxi kasutajad terminali armastavad.

Teises alapeatükis tuuakse näide, kus hea käsurea kasutamise oskus säästab meie aega. Näitena on kasutatud suurt kausta, mis koosneb suvaliste failinimedega piltidest, kuid kasutaja soovib pildid mõistlikult nummerada. Selle teostamiseks on toodud näide ühest võimalikust käsust, mis selle töö hetkega ära teeb (joonis 2).

```
ls -v | cat -n | while read n f; do mv -n "$f" "$n.jpg"; done
```

Vastava käsu jooksumisel muutuvad kõik .jpg failid failideks nimega 1.jpg, 2.jpg 3.jpg ... jne. Püstkriipsud tähistavad ühel real olevate erinevate käskude vahekohti. Täitmist alutatakse vasakpoolseimast käsust.

Joonis 2. Näide failinimedede muutmise käsust.

Järgnevalt seletatakse Linuxi tarkvara uuenduste toimimist. Uuendamiseks vajalikud käsud on selleks ajaks juba õpilasele tuttavad, kuid siin peatükis seletatakse lähemalt nende tähendust ning täpsemat eesmärki (joonis 3). Tuuakse välja terminali kaudu kõigile programmidele korraga toimiva tarkvarauuenduse häid külgi ning võrreldakse tarkvarauuendust Windowsiga.

Need kaks käsklust on võimalik asendada üherealise käsklusega: `sudo apt update -y | sudo apt upgrade -y`

Sudo - *superuser do* on programm, mis laseb kasutajatel jooksutada rakendusi lisaprivileegidega. Kui tegeletakse ülesannetega, mis vajavad superuseri õigusi, siis pannakse käsu ette sudo. Pärast käsu käivitamist küsitakse kasutaja parooli, et protsess oleks turvalisem. Käsureal parooli sisestamisega võib jääda tunne, et parooli ei saa sisestada. Tegelikult on lihtsalt parool silmade eest peidetud, et lisada turvalisust. Sellest käsust võib mõelda kui adminiõigustest, millega on tagatud suurem juurdepääs oma süsteemile.

Update - *update* käsuga laaditakse Sinu süsteemi saadaolevate pakettide **informatsioon** - selle käsu täitmisega ei toimu veel programmide uuendamist. See tähendab, et update käsuga salvestatakse /etc/apt/sources.list faili nimekiri pakettidest, mis on uuendamiseks saadaval.

Upgrade - *upgrade* aga uuendab ja installeerib nimekirja alusel uued paketid Sinu süsteemi.

Üks | märk käskude vahel tähendab seda, et esmalt täidetakse vasakul pool olev käsk ning kui see oli edukas, siis liigutakse märgist paremale jääva käsu täitmise juurde.

-y on parameeter. Nagu teame, saab käskudele juurde anda teatud parameetreid. Näiteks käsk **ls** kuvab jooksva kausta sisu, aga käsklus **ls -a** kuvab lisaks ka peidetud failid vaadeldavas kaustas. Antud käsklustes tähistab parameeter **-y** sõna "yes", ehk jah. Kui jooksutame käskluse **sudo apt update**, siis küsitakse meilt luba, kas soovime ikka pakettide nimekirja uuendada ning loa andmiseks peame sisestama tähe "y" ning vajutama klahvi "Enter". Parameetriga **-y** anname teada, et oleme uuendustega nõus ning luba eraldi ei küsita.

Joonis 3. Linuxi Apt tarkvara hallatavate pakettide uuendamise käsklused.

Esimese tunni viimases alapeatükis õpetatakse aliaste ehk otseteede loomist. Alguses seletatakse aliase eesmärki ning tuuakse näide selle rakendamisest. Ühtlasi räägitakse ajutistest ning püsivatest aliastest ning õpetatakse mõlemat iseseisvalt ka tegema. Materjal eeldab, et toodud näited teeb õpilane läbi paralleelselt lugemisega.

Teine tund käsitleb skripte. Tunni alguses seletatakse, mis on Bash skript, millele järgneb juhend esimese skripti loomiseks ning käivitamiseks. Esimene skript prindib käsureale tervituse "Tere", millele järgneb arvutikasutaja nimi. Kui esimene skript on tööle pandud, tehakse praktiliselt läbi sammud, mille tulemusel saab loodud skripti käivitada asudes ükskõik millises kaustas. See on oluline oskus, sest skripti kasutamiseks ühte kindlasse kausta liikumine igal kasutuskorral on üleliigne tegevus.

Järgneb skripti loomise täpsem õpetus. Kõigepealt tutvustatakse muutujate loomist ning kasutamist (joonis 4). Sellele järgneb väike ülesanne, kus tuleb luua skript, mis küsib kasutajalt vanust, elukohta ning lemmikfilmi ja väljastab kogutud informatsiooni ekraanile. Peale ülesannet õpetatakse loogikalausete ning tsüklite tööpõhimõtet näidisskriptide põhjal

(joonis 5). Materjalides ei eeldata, et õpilane oskab loetut peast kasutada, eesmärk on tutvustada õpilasele skripti põhikomponente. Seeläbi suudab õpilane mõista paremini ka keerulisemate skriptide sisu. Samuti oskab õpilane ise skripte luues mõelda, mida ta teab, et skriptides teha saab ning kust saab täpselt süntaksi üle vaadata.

Kasutades ülakomasid (') teksti ümber, ei asendata seal sees muutujate väärtusi (ehk märgil \$ ei ole mõju, kui see on ülakomade vahel)

```
echo 'Ei asenda muutujat $1 tema väärtusega'
```

Kui aga kasutada juttumärke ("), toimub asendus ilusti.

```
echo "Asendatakse muutuja $1 tema väärtusega"
```

Joonis 4. Muutujate kasutamine sõnedes.

While

While tsükli sisu korratakse seni, kuni tingimus on tõene. Näites sooritatakse järgnev sisu, kuni muutuja \$a on väiksem kümnest:

```
#!/bin/bash
a=0
while [ $a -lt 10 ]
do
    echo $a
    a=$((a+1))
done
```

For tsükkel

For kordus on kasulik mingi nimekirja või listi läbi vaatamiseks:

```
#!/bin/bash
a="Martin Janar Tiit Juku"
for i in $a
do
    echo "$i"
done
```

Joonis 5. Tsüklite kasutamine skriptis.

Eelduste kohaselt on õpilane eelneva materjali lugemise käigus proovinud ise erinevaid skripte luua ning käivitada. Selleks, et anda õpilastele veel paremat ülevaadet skriptide võimalustest, tuuakse näited keerulisematest skriptidest, mille on loonud igavpäevased Linuxi kasutajad. Mõlema näite puhul on õpilastel võimalik skripte mitte ainult lugeda, vaid ka praktiliselt käivitada ning testida. Esimene näiteskript võimaldab videojagamise keskkonna Youtube videosid otsida käsurealt ning seejärel käivitada video lokaalses meediamängijas.

Teine skript on “meemi generaator”, mille abil saab valida tuntumate meemide taustapiltide seast sobiva ning lisada pildile meelepärase teksti.

Tunni viimane osa koosneb peatükki hõlmavatest harjutustest, mis tuleb peale sooritamist kas õpetajale ette näidata või edastada õpetaja meiliaadressile. Harjutustes tuleb luua kaks meelepärast aliaast, seejärel kirjeldada käskluste “update” ja “upgrade” erinevust ning viimasena otsida internetist endale meelepärane skript, seda katsetada ning selle toimimist kirjeldada. Vastused tuleb salvestada tekstidokumenti.

3.3 Raspberry Pi materjalid

Kursuse teine, väiksem, osa sisaldab kuut 90-minutilist tundi, milles käsitletakse Raspberry Pidi. Teemade valimisel lähtuti ideest, et ükski teema ei tohiks muutuda õpilase jaoks üksluiseks ning õpilasel võiks peale tundi püsida huvi teema kohta rohkem uurida. Kursuse Raspberry Pi osa sisaldab mitmete erinevate riistvaraliideste kasutamist. Kursus lõpeb tunniga, kus õpilased saavad kirja panna oma projekti idee. Seeläbi saab õpilane julgustust jätkata Raspberry Piga ka peale kursust, et teostada kirja pandud projekt. Kursuse teise osa kuus teemat koos kirjeldava lausega:

1. Raspberry Pi tutvustus ja sissejuhatus. Lisaks Raspberry Pi tutvustamisele pannakse arvutile peale operatsioonisüsteem ning tehakse esmane käivitamine.
2. Raspberry Pi puutetundlik ekraan ja LED-tuled. Miniarvutile pannakse külge puutetundlik ekraan, seejärel valivad õpilased endale jõukohased ülesanded LED-tulede tööle panemiseks.
3. Raspberry Pi ja Sense HAT. Tunnis õpitakse Sense HATi LED ekraanile teksti ning mustreid kuvama, tunni lõpus saavad õpilased uurida koodi, millega saab LED ekraanil mängida populaarset mängu “Flappy bird”.
4. Raspberry Pi kaamera moodul. Tunnis õpitakse kaamera kasutamist nii käsurea kui ka Pythoni keeles programmeerimise abil. Tunni lõpuks valmib kiirematel õpilastel liikumisanduriga turvakaamera.

5. Raspberry Pi kasutamine eemalt (SSH ja VNC). Luuakse kaugühendus Ubuntu või Windowsi arvuti kaudu, et lülitada eemalt sisse LED-pirne või vaadata turvakaamera tehtud pilte.
6. Raspberry Pi oma projekt ja huvitavad näited. Õpilastele tutvustatakse erinevaid Raspberry Pi lahendusi ning seejärel valivad õpilased endale meelepärase projekti idee, mille täitmiseks kirjutatakse projekti teostamise plaan.

M. Tamme ülesanne oli luua õppematerjal esimesele ja kuuendale peatükile, bakalaureusetöö autor lõi materjalid teisele kuni viiendale peatükile. Materjalide loomise protsess toimus sarnaselt Linuxi materjalide loomisele.

4. Kursuse materjalide pilootimine

Kursust pilooditi ühes Eesti gümnaasiumis õppeaasta 2019/2020 teisel perioodil (28.10 - 13.12). Kursusele oli registreerunud 16 õpilast, kellest 15 läbisid kursuse edukalt. Üks õpilane jättis valikkursuse pooleli isiklikel põhjustel. Kõik kursusel osalenud õpilased võtsid esimesel perioodil programmeerimiskeeles Python toimunud programmeerimiskursust. Vastavalt kooli tunniplaanile toimusid tunnid:

- esmaspäeviti kell 8.00 - 09.30,
- kolmapäeviti kell 13.45 - 14.30,
- reedeti kell 13.45 - 15.15.

Seega toimus nädalas kaks 90-minutilist ning üks 45-minutiline tund. Selline korraldus tõi selgelt välja, et 45-minutilise tunni puhul ei jõua õpilased eriti hästi teemasse süveneda. Probleeme tekitasid ka kord nädalas 45-minutilise tunni tõttu tekkinud olukorrad, kus 90-minutilises tunnis võeti esimesed 45 minutit ühe peatüki teist tundi ning järgmised 45 minutit uue peatüki esimest tundi. Ühes tunnis kahe erineva teema käsitlemine tähendas õpilaste jaoks pidevalt teemade vahetumisel uuesti sisse elamist. Teiseks muutis see kohati raskemaks pilootimise olulist poolt – materjalide kvaliteedi hindamist.

4.1 Tunni korraldus ning õpetaja roll tunnis

Koolitunde viis läbi bakalaureusetöö autor. Mõnes tunnis oli kohal ka valminud materjalide kaasautor Marten Tamm, kes aitas õpilaste küsimustele vastamisega. Peaaegu kõigis tundides oli kohal kooli programmeerimisõpetaja, kes vastutas õpilaste lõpphinde väljapanemise eest, kuid ei osalenud aktiivselt tunni läbiviimises – tema täitis tundides vaatleja rolli.

Iga uue peatüki alguses tegi tunni läbiviija umbes viieminutilise sissejuhatuse tunni teemasse. Seejärel tegelesid õpilased iseseisvalt materjalidega. Küsimuste ja murede korral pöördusid õpilased abi saamiseks kas kaasõpilaste (mis oli tunni läbiviija poolt tugevalt julgustatud tegevus) või läbiviija poole. Õpilaste progressi jälgimiseks lõi töö autor tabeli, kuhu märgiti õpilase nime taha plussmärk, kui õpilase vastava teema ülesanded olid läbi vaadatud ning arvestatuks hinnatud. Kuna materjalide loomise käigus oli üheks püstitatud eesmärkidest võimalikult vähene vajadus õpetaja rollile tunnis, siis üritas töö autor õpilasi aidates suunata

materjalides õigest kohast lahendust leidma, mitte otseselt vastata õpilase küsimusele. See kandis kahte eesmärki – esiteks tulid selgelt välja materjalide miinuspooled, ehk teemad, mille kohta tekkis küsimusi paljudel õpilastel ning teiseks arendas selline viis õpilaste oskust probleeme iseseisvalt lahendada. Töö autor tegi tunni läbiviimise ajal märkmeid ning täitis eneseanalüüsi küsimustiku tuginedes märkmetele peale tunni toimumist.

4.1.1 Linuxi kontrolltöö

Linuxi osa kursusest lõppes kontrolltööga, mille koostas bakalaureusetöö autor. Kontrolltöö toimus spetsiaalselt ettevalmistatud virtuaalmasinas ning õpilased võisid vabalt kasutada kursuse materjale. Kontrolltöö ülesanded olid peamiselt praktilised, mis kontrollisid õpilaste õpitud oskusi. Paar ülesannet panid õpilase ka uude olukorda, millega kontrolliti õpilase võimekust probleeme lahendada. Kontrolltöö viimased kaks küsimust kontrollisid õpilaste kaasamõtlemit. Õpilastelt küsiti nende arvamust Linuxist ning mida nad oskavad öelda Linuxi turvalisuse kohta.

Kontrolltöö läbisid edukalt kõik töö sooritanud õpilased. Peaaegu kõik õpilased jõudsid pooleteise tunniga lahendada kõik kontrolltöö ülesanded. Kontrolltöö koostamisel loodi ülesandeid arvestusega, et vaid kiiremad jõuavad kõik ülesanded sooritatud. Viimase kahe küsimuse vastused osutusid väga heaks näiteks, et kursus äratas õpilastes huvi Linuxi vastu. Nimelt selgus vastustest, et mitu õpilast on ka kodus katsetanud Linuxi kasutamist. Samuti vastasid paljud õpilased küsimustele teadmistega, mida kursuse materjalides ei õpetatud. Arvestades, et kursuse alguses oli vaid üks õpilane tuttav Linuxi operatsioonisüsteemiga, on tulemus väga positiivne.

4.1.2 Raspberry Pi projekti teostamise plaan

Raspberry Pi teemad olid õpilaste jaoks kaasahaaravad ning pakkusid põnevust igal tasemel õpilasele. Mitmed õpilased tõid kursuse lõpus välja, et kavatsevad oma tunnis loodud projektiplaani iseseisvalt teostada. Projektiidee valimise tunnis oli näha, et õpilased valisid projekti põhjalikult, mõeldes oma huvidele. Kõik kuusteist õpilast valisid teostamiseks erineva projekti. Viimase ehk projektiplaani kirjutamise tunni põhjal saab öelda, et erinevate kasutusvõimaluste ning riistvara moodulite tutvustamine inspireeris õpilasi Raspberry Piga tegelema.

4.2 Pilootimise tulemuste hindamine

Kursuse pilootimise tulemuste dokumenteerimiseks otsustati luua eneseanalüüs tunni läbiviijale. Selleks koostas autor kümnest küsimusest koosneva küsimustiku (lisa IV), millele vastati iga tunni toimumise järel. Tunnis vaatleja rolli täitnud kooli õpetaja andis omapoolse tagasiside tunni läbiviija täidetud küsimustikule. Tagasiside küsimustiku eesmärk oli hinnata tunni korraldust ning õpilaste kaasatust tundi. Selline pilootimise tulemuste hindamise viis valiti koostöös juhendaja E. Tõnissoniga. Teise variandina kaaluti õpilastelt tagasiside küsimist. Tunni läbiviija on materjalid ise loonud ning tunneb neid põhjalikult, seega leiti, et eneseanalüüsi teel on võimalik tuvastada murekohti täpsemalt. Samuti oleks õpilastel väsitav iga peatüki lõpus täita põhjalik tagasiside ning vastused võivad kergesti muutuda pinnapealseteks.

4.2.1 Eneseanalüüsi vormi loomine

Tagasiside vormi loomisele eelnevalt uuris töö autor erinevaid teadustöodes või õpetajatele suunatud artiklites välja toodud küsimustikke. Peamiseks raskuseks kasutusele võetud küsimustiku loomisel osutus materjalide loomisel püstitatud eesmärk, kus kontakttunnis peaks olema õpetaja roll võimalikult väike, et kursuse õpetamine ei tugineks täielikult õpetaja teadmistele, vaid materjalides orienteeruv õpetaja saaks materjale tundides kasutada. Seega on püstitatud eesmärk kombinatsioon veebikursusele sarnanevast iseõppevormist ning koolitunnist. Leitud eneseanalüüsi küsimustikud keskendusid koolitundide puhul peamiselt õpetaja rollile tunnis. Veebikursustele mõeldud vormid keskendusid täielikult iseseisvale õppele. Seega otsustas töö autor eneseanalüüsi küsimustiku kokku panna tuginedes Hille Voolaidi ja Karel Pajuse koostatud dokumendi “Õppetunni läbiviimise meetodika ja õppetunni analüüs” lisas number 14 välja toodud õpetaja eneseanalüüsi küsimustikule. Küsimustikust valiti välja küsimused, mis olid töö autori arvates olulised loodud materjalide hindamiseks. Valitud küsimuste sõnastust muudeti vastavalt vajadusele, osale küsimustele lisati juurde täpsustavaid küsimusi. Näiteks küsimustikus esinevad küsimused “Kas kõik lapsed töötasid aktiivselt kaasa? Kuidas ma seda tean?” muudeti kujule “Kas kõik õpilased töötasid aktiivselt kaasa? Kas materjal tundus õpilastele põnev?”, kuna autor peab eduka kursuse aluseks kaasahaaravat sisu. Välja jäeti küsimused, mis ei olnud selle kursuse eesmärkide hulgas, näiteks küsimus “Kas iga laps sai tunnis minult tähelepanu?”, kuna sellise

küsimuse vastus ei ole oluline materjalide kvaliteedi hindamise juures. Üks selle töö autori lisatud küsimus on “Võttes arvesse eelnevaid vastuseid, millised muudatused tuleks sisse viia antud tunni materjalides?”. Selle küsimuse vastustele tuginedes viidi kiiresti sisse parandused materjalides ning kaardistati suurimaid murekohti.

4.2.2 Pilootimise tulemused eneseanalüüsi ning vaadelnud õpetaja tagasiside põhjal

Töö autori arvates täitis koostatud küsimustik oma eesmärgi. Küsimustiku koostamise käigus eeldati, et osade küsimuste vastused võivad kohati kattuda. Samas loodeti erineva nurga alt küsimuste abil tuvastada rohkem probleeme kui seda oleks võinud välja tuua täielikult mittekattuvad küsimused. Selline küsimuste ülesehitus raskendas küsimustele vastamise protsessi, kuna autoril tekkis tihti soov vastata järgnevale küsimusele eelmises vastuses olnud sisuga, näiteks tekitasid sellist olukorda küsimused “Kas iga õpilane saavutas midagi selle tunni jooksul?” ja “Kas iga õpilane tundis positiivseid emotsioone selle tunni jooksul?”. Sellise vastamise vältimiseks tuli autoril iga küsimuse juures põhjalikult mõelda läbi tunnis toimunu. Seetõttu usub autor, et tagasiside oli tugevalt mõtestatud.

Valminud eneseanalüüsi küsimustike tulemuste põhjal on näha, et koostatud materjalid olid õpilastele jõukohased ning valikkursuse korraldus oli meelepärane nii õpilastele kui ka tunnis vaatleja rollis osalenud kogunud õpetajale. Õpilastele meeldisid kõige rohkem peatükid, mille praktilised ülesanded võimaldasid loovust ning valikuvabadust. Näiteks võib eneseanalüüsi põhjal välja tuua, et peatükk “Linuxi töölaud” haaras õpilasi, kuna õpilased said oma soovi järgi kujundada operatsioonisüsteemi väljanägemist ning kasutatavat värviskeemi. Kõige raskemad olid õpilaste jaoks esimene käsirida puudutav tund ning Linuxi turvalisust puudutav tund, kus käsitleti kasutajate õigusi failipuus. Ükski tund ei ebaõnnestunud. Seega ei vajanud materjalid suuremas mahus ümbertegemist. Väiksemad tekkinud probleemid lahendati materjalides kursuse läbiviimise käigus. Näiteks vormistati eneseanalüüsi põhjal ümber õpilaste iseseisvaid ülesandeid, mis osutusid liiga mahukaks. Ühtlasi parandati peale tunni toimumist materjalides nende lõikude sõnastust, mille põhjal tekkis õpilastel enim küsimusi. Mitmete peatükkide juures muudeti teksti esiletõstmist, et olulisem sisu oleks paremini eristatav.

Õpilased said hästi hakkama Linuxi kontrolltööga. Kogu kursuse jooksul oli õpilaste osalus tunnis kooliõpetaja sõnul keskmisest kõrgem ning õpilased olid tundides motiveeritud kaasa töötama. Kooliõpetaja hinnangul on materjalid õpetamiseks mugavalt kasutatavad ning 2020. aasta kevade seisuga plaanib kooliõpetaja uuel õppeaastal valminud kursust õpetada. Võttes arvesse eelnevat saab öelda, et valminud materjalid on sobilikud valikkursuse korraldamiseks gümnaasiumis, kuna materjalid tekitasid õpilastes huvi õppida.

5. Kokkuvõte

Selle bakalaureusetöö eesmärk oli luua terviklik valikkursuse materjal gümnaasiumiastmele, mis käsitleks operatsioonisüsteemi Linux ning arvutit Raspberry Pi. Materjalide hindamise teostamiseks oli eesmärk pilootida kursust ühes Eesti gümnaasiumis.

Selle bakalaureusetöö autor koostas koostöös Marten Tammega kursuse materjalid 2019. aasta suvel. Sama aasta sügisel pilooditi kursust ühes Eesti gümnaasiumis. Pilootimisel oli tunnis juhendaja rollis bakalaureusetöö autor ning vaatleja rollis kooliõpetaja. Pilootimise tulemuste hindamiseks loodi eneseanalüüsi küsimustik, millele vastas bakalaureusetöö autor peale igat tundi. Täidetud küsimustikule andis täiendava tagasiside tunni vaatleja. Eneseanalüüsi põhjal tehti kursuse materjalide parandusi pilootimise käigus.

Pilootimise eduka õnnestumise põhjal võib öelda, et valminud on kasutatav õppematerjal, mis on õpilastele kaasahaarav. Valminud veebimaterjal on õpilastele sobiv allikas koolitunnis õppimiseks. Kontrolltöö tulemused näitasid, et õpilased omandasid kursusest teemakohaseid teadmisi ning oskasid Linuxit kasutada. Pilootimise käigus kogutud eneseanalüüsi vastuste põhjal saab järeldada, et kursuse läbiviimisel on veebimaterjal peamiseks infoallikaks ning õpetaja roll tunnis on teemad sisse juhatada ja vastata küsimustele. Autori hinnangul on kursus kasutatav õpetajale, kes on eelnevalt kursuse ise põhjalikult läbi töötanud ning oskab seeläbi aidata õpilasi tekkinud muredega. Pilootimisel vaatleja rollis olnud õpetaja kavatses 2020/2021 õppeaastal valminud materjalide põhjal valikkursust ise õpetada.

Valminud kursuse saab võtta tulevikus aluseks sarnaste kursuste loomiseks. Autori arvates võiks järgmise sammuna valmida kursus, mis käsitleks mitut erinevat Linuxi distributsiooni. Kursuse Linuxi peatükkides (valminud 2019. aasta suvel) kasutatakse distributsiooni Ubuntu versiooni 18.04, bakalaureusetöö kirjutamise ajal (2020. aasta aprilli teises pooles) toodi välja Ubuntu versioon 20.04. Seega on üks edasiarenduse võimalustest materjalide üleviimine uuele Ubuntu versioonile. Valminud kursus on suures osas läbitav ka uue Ubuntu versiooniga, esmajoones tuleks materjalides üle vaadata ekraanivaated. Autori arvates ei ole uue versiooni väljatuleku tõttu valminud materjalid kohe iganenud, kuna Ubuntu 18.04 tarkvarauuendused jätkuvad veel mitu aastat ning suur hulk inimesi jätkab vanemate versioonide kasutamist.

Viidatud kirjandus

- [1] Tartu Ülikooli statistika. <https://www.ut.ee/et/sisseastumine/statistika-0> (06.04.2020)
- [2] Rille K. Digipöördeprogramm 2020-2023.
https://www.hm.ee/sites/default/files/1_digiprogr_2020_23.pdf (12.04.2020)
- [3] Gümnaasiumi informaatika ainekava.
<https://www.hitsa.ee/ikt-haridus/progetiiger/gumnaasiumi-informaatika-ainekava>
(12.04.2020)
- [4] Torvalds L. LINUX's History. 1992. <https://www.cs.cmu.edu/~awb/linux.history.html>
(05.05.2020)
- [5] Free Software Movement. <https://www.gnu.org/> (01.12.2019)
- [6] Glossary of Linux, BSD and open source terms.
<https://distrowatch.com/dwres.php?resource=glossary#distribution> (01.12.2019)
- [7] Search Distributions.
https://distrowatch.com/search.php?ostype=All&category=All&origin=All&basedon=All&no_tbasedon=None&desktop=All&architecture=All&package=All&rolling=All&isosize=All&netinstall=All&language=All&defaultinit=All&status=Active#simple (01.12.2019)
- [8] What is Debian?. <https://help.ubuntu.com/lts/installation-guide/s390x/ch01s02.html>
(27.04.2020)
- [9] The story of Ubuntu. <https://ubuntu.com/about> (15.01.2020)
- [10] Vaughan-Nicholas S. J. What's the most popular Linux of them all? 2018.
<https://www.zdnet.com/article/whats-the-most-popular-linux-of-them-all/> (01.12.2019)

- [11] Usage statistics of Linux for websites. <https://w3techs.com/technologies/details/os-linux> (01.12.2019)
- [12] Moorhead J. Raspberry Pi device will 'reboot computing in schools'. The Guardian. 2012. <https://www.theguardian.com/education/2012/jan/09/raspberry-pi-computer-revolutionise-computing-schools> (06.05.2020)
- [13] Cambridge Raspberry Jam. https://camjam.me/?page_id=618 (24.04.2020)
- [14] Wolfer P. Peek - an animated GIF recorder. <https://github.com/phw/peek#about> (12.04.2020)
- [15] Teaduse populariseerimise alategevuse „Teeme+” toetuse andmise tingimused ja kord. RT I, 20.10.2017, 18. <https://www.riigiteataja.ee/akt/120102017018> (06.04.2020)
- [16] Operating System Market Share. <https://netmarketshare.com/operating-system-market-share> (06.05.2020)
- [17] Tartu Ülikooli informaatika bakalaureuseõppekava. https://www.cs.ut.ee/sites/default/files/www_ut/oppimine/infbak2019_2020.pdf (27.04.2020)
- [18] Tallinna Tehnikaülikooli informaatika õppekava. https://ois.ttu.ee/portal/page?_pageid=37,674560&_dad=portal&_schema=PORTAL&p_action=view&p_fk_str_yksus_id=50001&p_kava_versioon_id=50529&p_net=internet&p_lang=ET&p_rezhiim=0&p_mode=1&p_from= (27.04.2020)
- [19] Usage statistics of operating systems for websites. https://w3techs.com/technologies/overview/operating_system (08.04.2020)
- [20] Developer Survey Results 2019. <https://insights.stackoverflow.com/survey/2019> (12.04.2020)

Lisad

I. Kursuse materjalide loomise tööjaotus

Marten Tamm - T

Magnar Vares - V

Teema	Mustand	Puhtand	Viimistlus enne pilootimist	Tunni läbiviija pilootimisel	Parandused peale pilootimist
Sissejuhatus Linuxisse	V	V ja T	V	V	V
Linuxi töölaud	V	V ja T	V ja T	V	V
Käsurida 101	T	V ja T	V	V	V
Ubuntu tarkvara	T	V ja T	T	V	V
Käsurida 202	V	V ja T	V ja T	V	V
Failiõigused ja käsurea kordamine	V	V ja T	V	V	V
Turvalisus	V	V ja T	V	V	V
Windowsi programmid Linuxis ja töölaua keskkonnad	T	V ja T	V	V	V
Kordamine ja kuidas Linuxit oma arvutis tööle panna	T	V ja T	T	V	V
Linuxi Kontrolltöö	V	V ja T	V	V	V
Raspberry Pi tutvustus ja sissejuhatus	T	V ja T	T	V	V

Raspberry Pi puuetundlik ekraan ja LED tuled	V	V ja T	V	V	V
Raspberry Pi ja Sense HAT	V	V ja T	V	V	V
Raspberry Pi kaamera moodul	V	V ja T	V	V	V
Raspberry Pi kasutamine eemalt (SSH ja VNC)	V	V ja T	V	V	V
Raspberry Pi oma projekt ja huvitavad näited	T	V ja T	V	V	V

II. Valminud kursuse materjalid

Kursuse “Linux ja Raspberry Pi” materjalid asuvad aadressil:

<https://courses.cs.ut.ee/t/nooredkoodi/Linux/Linux>

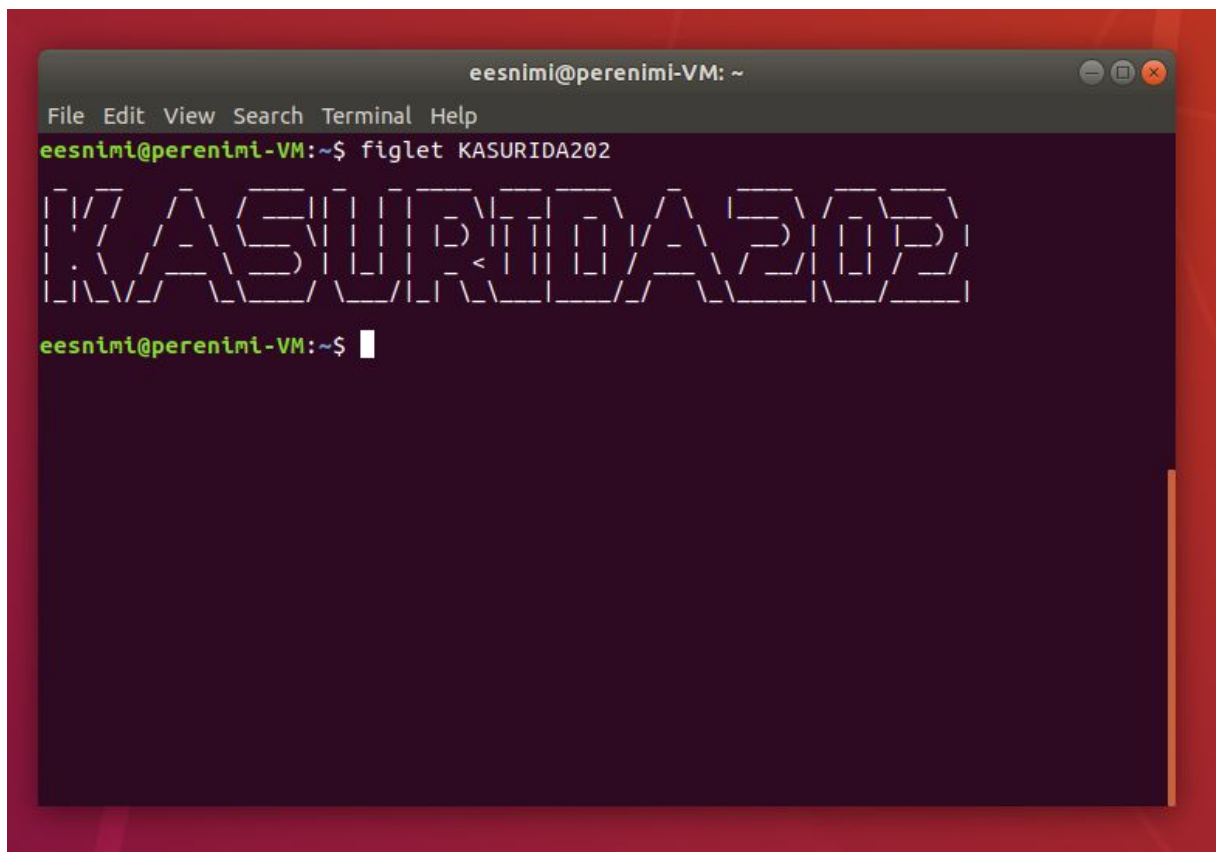
III. Tunni KÄSURIDA 202 materjal

Bakalaureusetöös näidistunnina kasutatud peatüki materjalid kursuse lehel:

<https://courses.cs.ut.ee/t/nooredkoodi/Linux/Kasurida202>

Käsurida 202 (Sissejuhatus tundi)

Nagu üleelmise praktikumi alguses rääkisime, on terminali kasutamine väga mugav ning muudab arvutis ringi käimise palju kiiremaks. Selles praktikumis aga hakkame vaatama erinevaid konkreetseid näiteid koos ise läbitegemisega. Käime läbi [Käsurida101 praktikumis](#) välja toodud punktid, et asjast paremini aru saada.



Esimene tund

1. Käsuriada aitab oma arvuti hingedel rohkem kursis olla

Kui Sa kasutad mõnda programmi koos kasutajaliidesega, siis kindlad nupud teevad kindlaid ülesandeid. Käsuriada teeb seda samuti, aga koos suurema täpsuse ja jõulisusega. Kui Sa kirjutad sõnu ja vajutad enter klahvi, siis käsuviip interpreteerib need sõnad endale arusaadavasse keelde ja töötab koos operatsioonisüsteemi kerneli ja failidega, et käivitada vastavaid käske. Terminali kasutades saab väga täpselt arvutile öelda, mida Sa teha soovid.

Räägime siia ühe loo, mis aitab selgitada käsurea vajalikkust ja lihtsust.

“Ma lõpuks saan aru, miks kogunenud kasutajad ainult terminalis ringi kolavad. See on nagu, kui Sa pead vanemale inimesele seletama, et kuidas lahti pakkida kompressitud fail ning liigutada faili sisu kuhugi kindlasse kausta. Kui kasutada kasutajaliidest mõnes ilusas programmis, siis on sellel lisaks nii palju erinevaid nuppe ja vidinaid, mis võivad valele teele viia. Ja kui avada Internet, et selle programmi kohta abi saada, siis võib tee peal kogemata klikkida mitmele reklaamile ja lõpetada lihtne Google'i otsing kahekümne uue extensioniga ja uhiuue otsimisribaga enne, kui õige infoni jõutakse.

See on midagi, mille saab kergesti lahendada terminaliga, sest Sa saad neile ette anda hulk käske, mis teevad kõik vajaliku tema eest ära. Ja kui tekib viga, millest aru ei saada, siis väga suure tõenäosusega leidub Internetis inimene, kellel oli probleem täpselt sama käsuga. Kiire otsinguga ja copy-paste'ga saad jätkata oma tööd sealt, kus pooleli jäi. See kõik hakkab vaikselt mulle kohale jõudma...”

allikas: Reddit r/Ubuntu “I finally understand why power users love the terminal.”

2. Sästab aega - mõned käsud aitavad hoida tohutult aega kokku

Failinimede muutmine

Meenuta, millal tõmbasid enda arvutisse näiteks terve hooaja sarju, mille pealkirjad olid pikad ning lohisevad või millal oli Sul arvutis sadu erinevaid allalaetud pilte, mille pealkirjad ei ole midagi muud kui üks pikk ja lohisev tähtederägastik

```
wi6xPGHoObhAhWLuIsKHxk2AVIQ_AUID.png
```

Kui aga Sul selliseid olukordi pole ette tulnud, siis ürita kaasa mõelda, kuidas käituksid Sina sellises olukorras. Oletame, et sul on kaustas Pildid 125 erinevat pildifaili. Igal failil on

pealkirjaks midagi arusaamatut. Kui tekib tahtmine oma kaustu organiseerida ning piltide pealkirjad muuta korrektseks, siis tuleb käsureal liikuda vastavasse kausta ning käivitada käsk

```
ls -v | cat -n | while read n f; do mv -n "$f" "$n.jpg"; done
```

Vastava käsu jooksumisel muutuvad kõik .jpg failid failideks nimega 1.jpg, 2.jpg 3.jpg ... jne. Püstkriipsud tähistavad ühel real olevate erinevate käskude vahekohti. Täitmist alustatakse vasakpoolseimast käsust.

Hoia oma süsteem hõlpsasti ajakohasena

Üheks suureks plussiks Linuxi puhul on uuendused. **Suur osa Sinu arvutis olevast tarkvarast uuendatakse automaatselt.** Siiski on olemas mõned uuendused, mis vajavad süsteemi restarti. Automaatselt värskendatakse selliseid rakendusi, mis on olulised turvalisuse pärast. Kui uuendada hakatakse sellist rakendust, mis parasjagu käimas on, siis oodatakse, kuni töö lõpetatakse. Kui programm uuesti avatakse, siis on uus versioon peale laaditud ning kasutamiseks valmis. Eelmises praktikumis õpitud Snap paketid on just need, mis ennast automaatselt uuendavad.

Siiski leidub rakendusi, mida tuleb ise uuendada. Arvatavasti oled juba kokku puutunud järgnevate käskudega:

```
sudo apt update
```

```
sudo apt upgrade
```

Need kaks käsklust on võimalik asendada üherealise käsklusega:

```
sudo apt update -y | sudo apt upgrade -y
```

Sudo - *superuser do* on programm, mis laseb kasutajatel jooksumata rakendusi lisaprivileegidega. Kui tegeletakse ülesannetega, mis vajavad superuseri õigusi, siis pannakse käsu ette sudo. Pärast käsu käivitamist küsitakse kasutaja parooli, et protsess oleks turvalisem. Käsureal parooli sisestamisega võib jääda tunne, et parooli ei saa sisestada. Tegelikult on lihtsalt parool silmade eest peidetud, et lisada turvalisust. Sellest käsust võib mõelda kui adminiõigustest, millega on tagatud suurem juurdepääs oma süsteemile.

Update - *update* käsuga laaditakse Sinu süsteemi saadaolevate pakettide **informatsioon** - selle käsu täitmisega ei toimu veel programmide uuendamist. See tähendab, et update käsuga salvestatakse `/etc/apt/sources.list` faili nimekirja pakettidest, mis on uuendamiseks saadaval.

Upgrade - *upgrade* aga uuendab ja installeerib nimekirja alusel uued paketid Sinu süsteemi.

Üks | märk käskude vahel tähendab seda, et esmalt täidetakse vasakul pool olev käsk ning kui see oli edukas, siis liigutakse märgist paremale jääva käsu täitmise juurde.

-y on parameeter. Nagu teame, saab käskudele juurde anda teatud parameetreid. Näiteks käsk **ls** kuvab jooksva kausta sisu, aga käsklus **ls -a** kuvab lisaks ka peidetud failid vaadeldavas kaustas. Antud käsklustes tähistab parameeter **-y** sõna "yes", ehk jah. Kui jooksutame käskluse **sudo apt update**, siis küsitakse meilt luba, kas soovime ikka pakettide nimekirja uuendada ning loa andmiseks peame sisestama tähe "y" ning vajutama klahvi "Enter". Parameetriga **-y** anname teada, et oleme uuendustega nõus ning luba eraldi ei küsita.

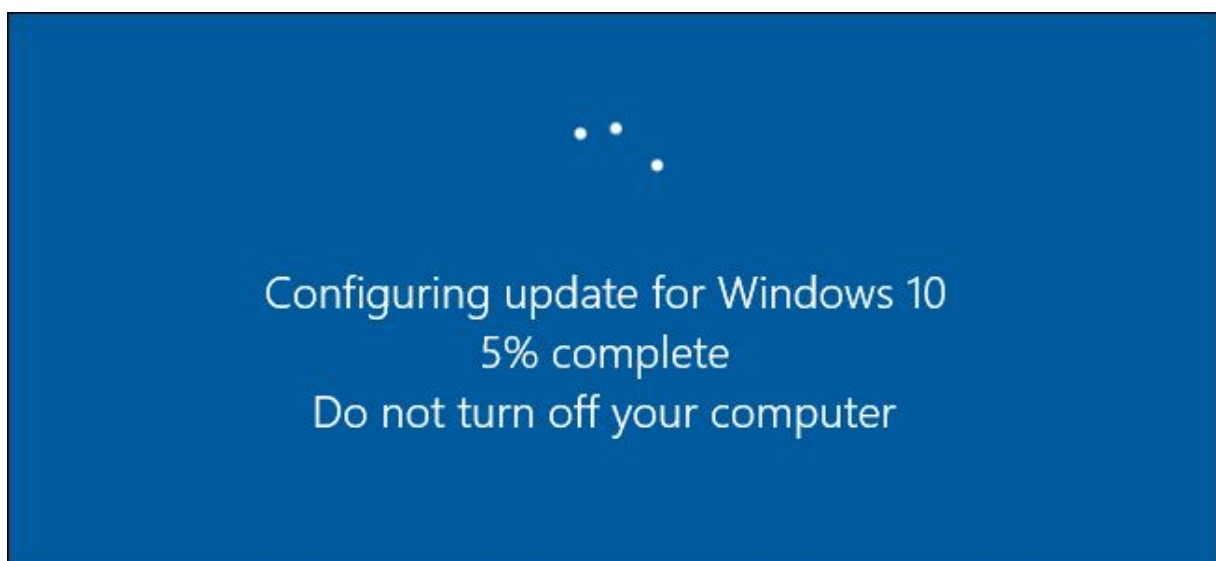
[Need kaks võivad küll vahepeal sassi minna, aga lihtne on mõelda:](#)

Update ehk *up to date* - süsteem paneb valmis nimekirja, mille põhjal programme installeerida saab.

Upgrade ehk *up the grade* - nimekirja põhjal installeeritakse uued paketid Sinu süsteemi.

Seda käsku tasub jooksutada aeg-ajalt, et hoida süsteem ajakohasena. Tegelikult teeb Ubuntu ise tihti päringuid ning uurib, kas mõni uuendus on saadaval. Vahel võid märgata, et ekraanile tekib aken, kus küsitakse, kas soovid alla laadida uuendused. Seal oma nõusoleku andmine on võrdne eelneva käsu jooksutamiselega terminaliaknas.

Windowsi kasutajatele teada-tuntud uuenduste ja restardi edasilükkamised on Linuxiga minevik.



3. Terminal on väga paindlik

Linuxi kasutajatel on tihti vaja jooksutada samu käske mitmeid kordi. Kirjutades või kopeerides samu käske ikka uuesti ja uuesti on ajakulukas ning tõmbab tähelepanu kõrvale olulisest tegevustest. Kogu raske töö saab säästa aliaستega, millega saab erinevaid käske kokku liita ja teha mitmeid otseteesid.

Juba süsteemis salvestatud otseteesid saame näha käsuga **alias**

Aliaseid saab koostada järgneva käsuga

```
@@alias otsetee_nimi="otsetee_käsk_jutumärkide_vahel"
```

Selle proovimiseks teeme aliase, et avada browser ja spotify.

```
alias pack="firefox | spotify"
```

Siin tuleb meeles pidada seda, et nii Firefox kui ka Spotify peavad selle näite jaoks olema installeeritud Sinu arvutisse. Võid ka proovida mõnda teist käsku.

See on lihtne ning kiire protsess, kuid peame meeles pidama, et on olemas kahte sorti otseteid ehk aliaseid - **püsivad ja ajutised**.

Ajutised otseteed ei salvestu Sinu süsteemi ning neid saab kasutada vaid terminalisessiooni jooksul. Kui avatud terminali aken sulgeda, siis on need otseteed kadunud. Ülevalpool tehtud pack alias on just ajutine.

Püsivad otseteed salvestatakse asukohta ~/.bashrc.

.bashrc on bash shelli skript, mis jooksutatakse iga kord, kui terminal avatakse. Seega sinna salvestatud otseteed aktiveeruvad alati, kui terminal lahti võetakse. Suures pildis on mõistlik salvestada otseteed alati püsivalt, et need iga terminaliaknaga oleksid uuesti käivitatud.

Selleks tuleb avada fail .bashrc, mis asub

```
/home/kasutajanimi/.bashrc
```

asukohas. Võime kasutada eelnevalt õpitud nano tekstiredaktorit koos käsuga **nano ~/.bashrc** või liikuda alguses kodukausta käsuga **cd ~** ning seejärel avada .bashrc fail.

Selles failis leia koht, kuhu oma edaspidiseid aliaseid salvestada. Otseselt ei ole vahet, mis koht selleks valida, aga soovitatav on hoida aliased ühes kohas, et kogu .bashrc fail oleks paremini struktureeritud.

Kui oled faili aliased juurde lisanud, salvesta fail ning terminalis jookseta järgnev käsk, et muudatused sisse viia **source ~/.bashrc** .

Source käsku on meil vaja käivitada selleks, et terminal uued muudatused kasutusele võtaks.

NB! Iga kord, kui avaneb terminal, käivitatakse esimese failina .bashrc, kust loetakse sisse kasutaja seaded (sealhulgas aliased ja muud seaded).

Aliaste kustutamiseks kasuta käsku **unalias aliase_nimi** (sessioonipõhiste aliaste kustutamiseks) või ava uuesti .bashrc fail ning kustuta soovitud rida ja salvesta.

Teine tund

4. Käsuriida aitab teha oma skripte, mis teevad elu palju lihtsamaks

Skriptimine

Kujuta ette mõne etenduse, filmi või telesarja skripti. Skript ütleb näitlejatele täpselt, mida ja kuidas teha, millal öelda põhjapanevaid lauseid, millal nutta ja millal naerda. Arvutile mõeldud skript ütleb arvutile samamoodi ette, mida ja kuidas teha ja kuidas käituda. Praeguses kontekstis keskendume me Bash skriptidele ja vaatame, mis on nende eelised ja kuidas skriptimisega algust teha.

Bash skript on tavaline tekstifail, mis koosneb erinevatest käskudest. Need käsud on samasugused nagu me oleme harjunud käsureale sisestama (nagu näiteks cp ja ls). Tuleb tähele panna, et kõike, mida me saame käivitada käsureal, saame me jooksetada skriptina ja tulemus on täpselt sama.

Skriptide eeliseks on kiirus ja lihtsus - nendega on võimalik automatiseerida väga mitmeid erinevaid tegevusi ning teha oma elu kordades kergemaks. Allpool vaatame näiteid, kuidas inimesed on teinud lühikesi skripte, et oma aega optimiseerida ja enda elu lihtsamaks teha.

Alustame ise oma esimese skripti loomise ja käivitamisega

Bash skripti päiseks ehk esimeseks reaks on alati **#!/bin/bash** Sellega määratakse skripti interpretaatoriks standardne POSIX-ühilduv käsuinterpretaator (Bourne Shell).

Käsureaskriptide kokkuleppeline failide laiend on `.sh`, aga palju kasutatakse ka lihtsalt ilma laiendita failinimesid - oluline on, et fail oleks käivitav (executable).

Võtame läbitegemiseks ette ühe lühikese skripti:

1. Loo uue kausta nimega "skriptid" oma kodukausta.
2. Liigume sellesse kausta ning loome uue faili `kursus.sh` kasutades nano tekstiredaktorit.

Sisuks lisa bash skripti päis ning *echo* rida:

```
#!/bin/bash
echo "Tere, "$USER
```

echo on käsuinterpretaatori käsk, mis väljastab ekraanile (standardväljundisse) oma argumentid.

3. Nüüd käivitame selle käsuga **sh kursus.sh**.

Kui käivitamisel näeme, et meil ei ole piisavalt õigusi, siis peame andma endale õigused seda skripti käivitada. Selle teema juurde tuleme tagasi järgmises praktikumis, aga seni piirdume vaid sellega, et peame käsureale sisestama järgmise käsu.

```
sudo chmod +x skripti_nimi.sh
```

4. Olemegi valmis! Nüüd saame käivitada oma tehtud skripti!

Käivita skripte igalt poolt

Kui aga soovime, et meie loodud skripte saaks käivitada igal pool, mitte vaid `~/skriptid` kaustas, siis tuleb anda süsteemile ette tee, kust ta skripte otsib.

Esmalt peame andma soovitatavatele skriptidele kõrgemad käivitamise õigused:

```
sudo chmod +x skripti_nimi.sh
```

NB! Soovitame selle Sul kindlasti läbi teha, kuna hiljem ei pea mõtlema, kuhu üks või teine skript salvestatud oli.

1. Liigu kodukausta ning ava fail “.bashrc”

```
nano .bashrc
```

2. Kirjuta uuele reale

```
export PATH=~/.skriptid:$PATH
```

Ning salvesta see fail.

Terminali uuesti avades viiakse sisse muudatused ning nüüd saame ~/.skriptid kaustas olevaid skripte käivitada igalt poolt.

Skripti käivitamiseks pole enam tarvis käsiklust **sh**, vaid piisab lihtalt skripti failinime kirjutamisest. Näiteks eelloodud skripti saame käivitada suvalisest kaustast käsuga **kursus.sh**.

Muutujad ja käsurea parameetrid

Skriptides on võimalik kasutada mitut erinevat tüüpi muutujaid:

1. Protsessilt protsessile päranduvad keskkonnamuutujad

\$PATH - kust otsida käivitatavaid programme, kui nende asukoht pole täisteega määratud

\$USER - kasutajanimi

\$PWD - protsessi hetkel aktiivne töökataloog

2. Skripti sisendi muutujad, mille kaudu saab ligi skripti argumentidele ja muule skriptiga seotud informatsioonile

\$0 - skripti nimi

\$1 \$2 .. \$9 - skripti käsurea argumendid

3. Skripti sisesed lokaalsed kasutaja poolt määratud muutujad

nimi="Juku" - Muutuja väärtustamisel piisab vaid nime määramisest (NB! Võrdusmärgi ees ja järel ei tohi olla tühikuid!)

echo \$nimi Muutuja väärtuse kasutamisel tuleb nime ette panna \$ märk:

Kasutades ülakomasid (,) teksti ümber, ei asendata seal sees muutujate väärtusi (ehk märgil \$ ei ole mõju, kui see on ülakomade vahel)

echo 'Ei asenda muutajat \$1 tema väärtusega'

Kui aga kasutada juttumärke ("), toimub asendus ilusti.

echo "Asendatakse muutuja \$1 tema väärtusega"

Interaktiivne sisend

Selleks, et küsida kasutaja käest täiendavat informatsiooni (näiteks failinime), saab kasutada Read käsku nii:

```
echo "Sisesta väärtus:"  
read sisse  
echo "Sisestati: $sisse"
```

Ülesanne

Muuda oma kursus.sh skripti nii, et küsitakse kasutaja käest üksikshaaval tema vanus, elukoht ning lemmikfilm. Sisestatud vastused tuleb väljastada ekraanile.

Tingimuslaused ja tsüklid

Sarnaselt muudele programmeerimiskeeltele on võimalik skriptides kasutada erinevaid tingimusi ja tsükleid.

If ja else

Teistest programmeerimiskeeltest juba hästi tuntud IF-ELSE konstruktsioon:

NB! Pöörake tähelepanu [] sulgude juures enne ja pärast tühikute kasutamisele!

```
#!/bin/bash  
a=$1  
if [ $a -lt 1000 ]  
then  
    echo "Sisestatud arv on vaiksem kui 1000"  
elif [ $a -ge 1000 -a $a -le 5000 ]  
then  
    echo "Sisestatud arv on vahemikus 1000 kuni 5000"  
else
```

```
fi echo "Sisestatud arv on suurem kui 5000"
```

Operaatorid teksti (stringide - kõik muutujad on siin stringitüüpi) võrdlemisel:

= - võrdne

!= - mittevõrdne

-z - On tühi

-n - Ei ole tühi

Operaatorid arvude võrdlemisel (stringide intepreteerimine tekstina esitatud arvudena):

-eq - võrdne (ingl. k. equal)

-ne - mittevõrdne (ingl. k. not equal)

-lt - väiksem (ingl. k. less than)

-le - väiksem või võrdne (ingl. k. less or equal)

-gt - suurem (ingl. k. greater than)

-ge - suurem või võrdne (ingl. k. greater or equal)

Ning loogikaavaldised:

-a (AND) nõuab mõlema tingimuse samaaegset täidetust

-o (OR) väljendab, et täidetud võib olla üks või teine või mõlemad.

! (NOT) alustades tingimust hüüumärgiga, märgitakse eitust

While

While tsükli sisu korratakse seni, kuni tingimus on tõene. Näites sooritatakse järgnev sisu, kuni muutuja **\$a** on väiksem kümnest:

```
#!/bin/bash
a=0
while [ $a -lt 10 ]
do
    echo $a
    a=$(( $a + 1 ))
done
```

For tsükkel

For kordus on kasulik mingi nimekirja või listi läbi vaatamiseks:

```
#!/bin/bash
a="Martin Janar Tiit Juku"
for i in $a
do
    echo "$i"
done
```

NB! Kui tegemist on teksti sisaldavate muutujatega, siis on soovitatav muutuja kasutamisel ümbritseda see jutumärkidega nii: **echo "\$i"**, sest siis võib kindel olla, et *bash* ei tõlgenda sõnet kui mõnda käsku. Kui eelmises nimekirjas oleks *Janari* asemel *echo*, siis saaks *bash* aru, et soovitakse käivitada *echo* käsku.

Kommentaariid lisavad skriptile/koodile arusaadavust ning on hea tava kommenteerida oma skripte, et kood oleks teiste jaoks loetav ning mõistev.

Kommentaare tähistatakse trellide #

YouTube terminalis

Kasutame järgmist skripti, mis laseb meil otse terminalist käivitada YouTube'i videosid.

Link repositooriumile on [siin](#).

Skripti saad kopeerida järgnevalt lehelt: [Skript](#)

Järgi samu samme, mis eelneva **kursus.sh** skripti puhul ning salvesta see nimega **yview.sh**

Kui õigused on antud, siis saame käsurealt otsida ja vaadata oma lemmikvideosid!

Video otsimiseks sisesta skripti nimi (nagu senini seda teinud oleme), seejärel tühikuga eraldatult märksonad, mis videosid otsida tahad. Näiteks käskluse **yview.sh funny dogs** käivitamisel kuvatakse valikut lõbusate koerte videodest.

NB! Kui Sul on arvutis VLC videomängija ning skriptis video valimise järel avanes küll VLC, kuid video esitamist ei toimunud, siis toimi järgnevalt:

1. Lae alla mpv videomängija järgmiste käskudega:

```
sudo add-apt-repository ppa:mc3man/mpv-tests
```

```
sudo apt update
```

```
sudo apt install mpv
```

2. Veendu, et mpv videomängija on sinu arvutis olemas.

3. Lae alla ning käivita modifitseeritud [ytview](#).

Miks VLC ei töötanud? Teatud VLC versioonides oi toimi Youtube'i videode mängimine. Selle mure lahendamiseks pakuvad internetifoorumid paari erinevat lahendust. Kuna pole kindlalt toimivat lahendust, mis laseks VLC kaudu Youtube'i videosi esitada, ei hakka meie neid kasutama.

mpv on videomängija, mille kaudu saab selle skripti abil samuti videosid vaadata. Kolmandas punktis toodud modifitseeritud skripti on muudetud nii, et skript ei otsiks arvutist VLC videomängijat, vaid kasutaks mpv videomängijat, kui see on arvutis olemas. Uuri ning võrdle esialgset ja modifitseeritud skripti või proovi ise esialgsest skriptist VLC videomängija valimine eemaldada.

Meme generator

Selle skriptiga saame koostada meeme käsurealt.

Link repositooriumile on [siin](#).

Skripti saad kopeerida järgnevalt lehelt: [Skript](#)

Salvesta see nimega **mime.sh**, anna õigused ning hakka koostama naljakaid meeme!

Meemide tegemiseks käivita terminalis **mime.sh** ning jälgi juhiseid

Terminalis saab pilte vaadata käsuga **cat mime.png**

Kui soovid pildi salvestada kindla nimega, siis kasuta skripti koos lisaparaameetriga -f, mis laseb failile oma nime panna **meme.sh -f omanimi**.

Teisi paraameetreid näeb käsuga **meme.sh -help**

Meemide malle ehk *template*'e näeb sellel [lehel](#), kus kaustade nimed tähistavad erinevaid malle.

Lisahuvi korral saab skriptide kohta lugeda ning erinevaid põnevaid koodijuppe läbi teha [siin](#)

Harjutused:

1. Ava uus tekstidokument nimega **vastused_käsurida202_OMANIMI.txt**, kus OMANIMI asemel lisad oma nime.
2. Loo kaks otseteed, millega muuta oma elu kergemaks ning lisa need tekstidokumenti. Abi leiad Internetist.
3. Kirjelda oma sõnadega, mis on *upgrade* ja *update* käsu vahe.
4. Otsi Internetist üks huvitav ning kasulik skript. Lisa see enda süsteemi ning kirjelda, mis see teeb.
5. Saada tekstifail juhendajale või näita seda talle.

Kokkuvõtte peamistest käskudest

\$ pwd - näita, kus kohas ma asun (print working directory)

\$ cd - vaheta asukohta (change directory)

\$ ls - näita kausta sisu (list)

\$ touch - loob uue faili

\$ file - näitab, mis failiga on tegu

\$ cat - loe (mitme) faili sisu (concatenate)

\$ less - loe mugavalt suurema faili sisu

\$ history - näitab sisestatud käskude ajalugu

\$ cp kopeerib faili etteantud teele

\$ mv - saame faile kas lõigata või ühest kohast teise tõsta

\$ mkdir kaustanimi - loo uus kaust

\$ rm - kusuta faile ja kaustu

\$ help - näitab käsu infot

\$ man - näitab käsu juhendit

\$ whatis - näitab, mida käsk teeb

IV. Eneseanalüüsi küsimustik tunni “Linuxi töölaud” näitel

Küsimus	Tunni läbiviija vastus	Vaatleja kommentaar vastusele (valikuline)
Kas kõik õpilased töötasid aktiivselt kaasa? Kas materjal tundus neile põnev?	Jah, kõik õpilased töötasid kaasa. Tegu oli väga põneva teemaga, sest õpilased said operatsioonisüsteemi välimust enda soovide järgi kujundada.	Nõus
Mis oli õpetaja roll tunnis?	Tunni alguses rääkisin veelkord üle, mida olime senini teinud. Andsin soovitusi, kuidas virtuaalmasinat korrektselt välja lülitada ning seletasin veel üht-teist virtuaalmasina loogika kohta. Enne tunni algust arutasid paar õpilast, kellest ühel oli kaasas USB-pulk Ubuntu installeerimisfailidega, kuidas see OS kooliarvutisse paigaldada. Peatasin selle tegevuse ning teavitasin ka tunni alguses kõiki õpilasi, et igasugused säärased muudatused ei kuulu selle kursuse sisu alla ning need tuleb kooskõlastada kooli õpetajatega.	Hea, et juhatad uue tunni sisse nii, et räägid, et kuhu oleme jõudnud
Kas oli õpilasi, kellele tundus töö liiga raske või kes ei saanud aru, mida tegema peab?	Mitmed õpilased kogesid tunnis hetki, kus nad täpselt ei osanud edasi tegutseda. Küsimus pole materjali puudulikkuses, vaid need kohad olid ette teada. Peamiselt oli probleemiks uue kausta loomine oma	Nõus

	<p>kodukausta- tegevus, mida iga õpilane on kindlasti Windowsi arvutis teinud ning samamoodi toimub see tegelikult ka Ubuntu. Pigem olen üllatunud, et Ubuntu-spetsiifilised tegevused, näiteks käsuraal mitme olulise käsu sisestamine, ei tekitanud kellelegi raskusi.</p>	
<p>Kas iga õpilane saavutas midagi selle tunni jooksul?</p>	<p>Jah. Iga õpilane muutis oma arvutis kuvatavate ikoonide kujundust ning töölaua üldist välimust. Ühtlasi panid kõik õpilased operatsioonisüsteemis kuvatava kella kõrvale ilmateate jne.</p>	<p>Nõus</p>
<p>Kas iga õpilane tundis positiivseid emotsioone selle tunni jooksul?</p>	<p>Arvan, et tegu oli ühe kõige põnevama tunniga kursuse jooksul, sest selles tunnis oli palju vabadust endale meeldivate kujunduste valimiseks. Kindlasti kogesid kõik õpilased positiivseid emotsioone.</p>	<p>Nõus</p>
<p>Kas distsipliiniprobleeme tekkis? Kas seda põhjustas: a) ülesannete keerukus b) halb sõnastus</p>	<p>Ei tekkinud.</p>	<p>Nõus</p>
<p>Kas ajaline planeering oli realistlik ja sobiv?</p>	<p>Tervikpilti vaadates pole ma kindel, kas materjal mahub täpselt 90 minuti sisse, kuid arvestades, et esimene osa (Linuxi installeerimine) läheb üldjuhul kiiremini, siis arvan, et planeering on sobiv.</p>	<p>analüüsid pädevalt</p>

Kas lugemine ning praktiline läbitegemine olid tunnis kaalus?	Jah, peaaegu terve tunni sai tegevusi jooksvalt kaasa teha.	Nõus
Millega olen väga rahul selles tunnis?	Mulle tundub, et selle teema asukoht kursuse materjalides on väga hea koha peal, sest see tund kindlasti tõstab õpilaste huvi Linuxi vastu.	Nõus
Võttes arvesse eelnevaid vastuseid, millised muudatused tuleks sisse viia antud tunni materjalides?	Iseseisva ülesande all tuleks sisse viia paar muudatust. Esiteks on palutud kirjeldada seadetest kahte iseseisvalt tehtud muudatust, kuid sõnastus ei keela välja tuua praktikumitöö käigus juhendi järgi tehtud muudatusi. Teiseks võib mahu vähendamise mõttes ümber sõnastada viienda punkti. Antud punktis palutakse laadida alla kaks uut kujundusteemat - üks ikoonide ning teine programmide jaoks. Võiks ette anda ühe konkreetse teema programmide jaoks.	
<p>Vaatleja üldine kommentaar: Jälgida, et kõik kuulaks kui kõneled.</p>		

V. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, **Magnar Vares**,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose

Valikkursuse „Linux ja Raspberry Pi” loomine ja pilootimine

mille juhendajad on Alo Peets ja Eno Tõnisson,

reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Magnar Vares
30.04.2020