

ŠIAULIŲ UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

Mindaugas Gapšys

Informatikos specialybės IV kurso dieninio skyriaus studentas

Bash skriptų panaudojimas Unix/Linux operacinių sistemų valdymui
BAKALAURO DARBAS

Darbo vadovas:
Doc. dr. L. Kaklauskas
Recenzentas:
Doc. dr. V. Giedrimas

Šiauliai, 2013

„Tvirtinu, jog darbe pateikta medžiaga nėra nuplagijuota ir paruošta naudojant literatūros sąraše pateiktus informacinius šaltinius, bei savo tyrimų duomenis“

Mindaugas Gapšys

.....
(parašas)

Darbo tikslai ir uždaviniai

Šio darbo tikslas – naudojant *Bash* skriptus parengti *Linux/Unix* operacinių sistemų valdymo programą. Tikslo įgyvendinimui spręsimė tokius uždavinius:

1. Teorinės medžiagos analizė, siekiant įvertinti *Bash* skriptų galimybes;
2. Suprojektuoti *Linux/Unix* operacinės sistemos valdymo modelį, atsižvelgiant į *Bash* skriptų galimybes;
3. Parinkti *Linux/Unix* operacinę sistemą ir parengti veikiantį kodą;
4. Ištestuoti operacinės sistemos valdymo programą.

Darbo vadovas
(vardas, pavardė) (parašas)

Turinys

Įvadas	5
1. TEORINĖ DALIS	6
1.1 Temos analizė	6
1.2 Darbo srities analizė	7
2. PROJEKTINĖ DARBO DALIS	8
2.1 Įrankių ir priemonių pasirinkimo analizė	8
2.2 Projekto (darbo) vykdymo planas	9
2.3 Pradinis projekto aprašymas	9
3. DARBO EIGOS APRAŠYMAS	12
3.1 Darbo eigos grafas	12
3.2 Problemų ir jų sprendimų aprašymai ir pagrindimai	13
3.3 Galutinio projekto būvio aprašymas	14
3.4 Darbo rezultatų analizė	15
3.4.1. Vartotojo sąsaja	15
3.4.2. Testavimas	17
3.5. Programos naudojimo sritis	20
3.6. Patarimai, pastebėjimai, rekomendacijos	21
IŠVADOS	22
Literatūros ir informacinių šaltinių sąrašas	23
Anotacija (Summary)	25
Priedai	26

Lentelių turinys

Lentelė 1 Darbų įvykdymo eiga	9
-------------------------------------	---

Paveikslėlių turinys

Pav. 1 Algoritmas pagal kurį veikia programa	11
Pav. 2 Algoritmas kaip buvo vykdomi darbai	12
Pav. 3 Linux apvalkalų sąsaja su vartotoju	14
Pav. 4 Iškvietus skriptą, vartotojui pateikiamas pasirinkimo meniu	14
Pav. 5 Iškvietus skriptą, administratoriui pateikiamas pasirinkimo meniu	15
Pav. 6 Panaudos atvejų diagrama (užduočių diagrama) (angl. use case diagram)	16
Pav. 7 Sekos diagrama (angl. sequence diagram), pagal kurią programa veikia	17
Pav. 8 Komandos <i>find</i> naudojimo pavyzdys	18
Pav. 9 Komandos <i>grep</i> naudojimo pavyzdys	19
Pav. 10 Pavyzdys, kaip veikia ps komanda	19
Pav. 11 Darbo kokybės užtikrinimo modelis (Wagenaar, 2008)	20

Ivadas

Linux operacinės sistemos šerdis – branduolys, kuris reguliuoja visus OS resursus. Tai yra duomenų išsaugojimas, spausdinimas, atminties ir failų tvarkymas. *Linux* apvalkalas – tai komandų įvedimo aplinka skirta vartotojui. Apvalkalas – speciali programa, kitaip komandų interpretatorius. Apvalkalas interpretuoja ir vykdo įvedamas komandas į programą, dažniausiai iš klaviatūros arba failo. Populiarėjantis *Linux* yra naudojamas asmeniniuose kompiuteriuose, superkompiuteriuose, o labai dažnai ir serveriuose *Linux* distribucijos (*Debian*, *RedHat*). *Linux* gali naudoti kiekvienas, nes yra nemokama (atviro kodo) operacinė sistema. *Linux*’e galima naudoti *Windows* programas pasitelkus *wine* programą. *Linux/Unix* valdymas vyksta dialogo pagalba (terminale). Pagrindinės komandos sutampa su *Windows* komandų sąrašų. Populiariausi *Linux* OS apvalkalai yra *Bash* (Bourne – Again Shell), *Csh* (C Shell) ir *Ksh* (Korn Shell). Bet kuris iš šių trijų apvalkalų vartotojo komandas priima iš klaviatūros arba failo. Tokie apvalkalai dar vadinami komandinės eilutės sąsaja (angl. interface). Norint su *Bash* parašyti elementarius skriptus, nereikia diegti jokių programų tik tekstiniam redaktoriuje parašyti specialią eilutę ir galūnę.

Kai kuriose *Unix* operacinėse sistemose (*Solaris*, *OpenBSD*, *IRIX*), norint naudoti *Bash* skriptus, reikia įsidiegti *Bash* paketus (angl. package), kurių pagalba būtų galima rašyti *Bash* skriptus.

Šio darbo tikslas – naudojant *Bash* skriptus parengti *Linux/Unix* operacinių sistemų valdymo programą. Tikslu įgyvendinimui spręsimė tokius uždavinius:

1. Teorinės medžiagos analizė, siekiant įvertinti *Bash* skriptų galimybes;
2. Suprojektuoti *Linux/Unix* operacinės sistemos valdymo modelį, atsižvelgiant į *Bash* skriptų galimybes;
3. Parinkti *Linux/Unix* operacinę sistemą ir parengti veikiantį kodą;
4. Ištestuoti operacinės sistemos valdymo programą.

1. TEORINĖ DALIS

1.1 Temos analizė

Skriptai yra nedidelės programos, sujungiančios skirtingus terpės komponentus į visumą. Sukurtus skriptus interpretuoja operacinės sistemos komandinės eilutės interpretatorius. Kiekvienas scenarijus yra tekstinis dokumentas, kuriame nurodomas veiksmų sąrašas, kas turi būti įvykdyta. Skriptai dažniausiai naudojami failų valdymui (ieškojimui, trynimui), programų vykdymui ir teksto spausdinimui. Apskritai, skriptų kalba yra lengviau ir greičiau programuojama nei labiau struktūriška ir išbaigta kalba, pvz.: C/C++, bet C/C++ parašytos programos gali būti įvykdytos net 2 kartus greičiau [17].

1991 m. suomių studentas Linus Torvalds, atlikdamas projektą, sukūrė naują, nemokamą, atvirojo kodo operacinės sistemos branduolį, žinomą *Linux* pavadinimu. *Linux* bendrai vadinama visa *Unix* operacinė sistema, naudojanti *Linux* branduolį kartu su sisteminėmis programomis bei bibliotekomis iš *GNU* projekto. Populiariausias *GNU/Linux* variantas yra *Ubuntu Linux*. Tai platinamasis paketas, kaip ir *OpenSuse Linux*, *Linux Mint*, *Fedora/Redhat*, *Debian GNU/Linux*, *Mandriva Linux*, *PCLinuxOs*, kurie gali būti taikomi specifiniams tikslams, tačiau populiariausi paketai skirti plačiajai vartotojų auditorijai [10]. Daugumoje distribucijos gali būti skirstomos pagal šiuos kriterijus [11, 14]:

- Namų vartotojui (angl. Desktop) dažniausiai darbalaukis su grafine sąsaja ir bendromis programomis (*Debian*, *Ubuntu*, *Fedora*, *OpenSUSE*, *Mandriva*, *Gentoo*, *Knopixx*, *Slackware*);
- Serveriams / Įmonėms (angl. Server / Enterprise) sukurta siekiant suteikti kelių vartotojų platformą verslui ir tinklinėms programoms naudoti. Tokių operacinių sistemų dėmesys orientuotas teikti paslaugas keliems klientams, o ne vieno vartotojo sąsajai (*Debian*, *Ubuntu*, *CentOS*, *OpenSUSE*, *Mandriva*, *RedHat*, *Gentoo*, *Slackware*);
- *Live*, kai operacinė sistema vykdoma tiesiai iš atminties laikmenų: USB, CD ar DVD. Ji skirta išbandyti ir pademonstruoti papildomas *Linux* galimybes (*Ubuntu*, *OpenSUSE*, *Knoppix*);
- Internetiniai nešiojamieji kompiuteriai (angl. Netbooks) šiems kompiuteriams pradėta leisti paprastesnės (angl. light-weight) GNU/Linux distribucijos (*Ubuntu*).

1.2 Darbo srities analizė

Bash yra nemokamas *Unix* apvalkalas, sukurtas Brajeno Fokso (Brian Fox). 1989 m. pradėtas platinti kaip *GNU* operacinės sistemos apvalkalas ir numatytasis komandų apvalkalas *Linux* operacinėje sistemoje, taip pat *MacOS X*. *Bash* galima naudoti *Microsoft Windows*, įdiegus posistemę *Unix* komandoms arba POSIX emuliatorių „Cygwin“ ar „MSYS“. Taip pat pritaikytas naudoti su *DOS* ir *Android* panaudojant emuliatorius. *Bash* yra komandų procesorius, paleidžiamas terminalo pagalba. Vartotojas gali įvesti komandas, kurios sąlygoja veiksmus. *Bash* sugeba perskaityti komandas iš failų. Slaptažodžiai, sintaksė ir kiti pagrindiniai kalbos bruožai buvo nukopijuoti nuo *sh* [3, 4, 12].

Bash skriptų pagalba galima beveik viską programiškai valdyti. *Bash* komandos internete yra pateiktos abecėlės tvarka su aprašymais. Pasirinktos komandos aprašyme yra paminėtos kitos susijusios komandos. Pradedančiajam vartotojui pradėjus naudotis *Linux/Unix* atrodo sudėtinga, pirmiausia dėl pasikeitusios grafinės aplinkos, kurios būna skirtingos, todėl, norint panašios aplinkos į *Windows*, rekomenduojama KDE aplinka. Norint valdyti *Linux/Unix* bus sudėtinga, nes komandos beveik visos skiriasi nuo *Windows* ir reikia atsiminti daug įvairių raktų ir kitos informacijos.

Grupavimo kriterijai nustatyti remiantis interneto resursų [8, 9] analize. Papildomai buvo peržiūrėti *Linux/Unix* atviri forumai pradedantiesiems, pažengusiems ir tinklo administratoriams siekiant patvirtinti pagal [8, 9] literatūros šaltinius atrinktus kriterijus. Suformuoti šeši komandų grupavimo kriterijai:

- Komandos skirtos peržiūrėti sistemos būseną;
- Darbas su failais, paieška failų;
- Darbas su katalogais;
- Vartotoju administravimas;
- Komandos peržiūrėti tinklo parametrus, bei valdymą;
- Paprastos, naudojamos dažniausiai be parametrų, komandos.

2. PROJEK TINĖ DARBO DALIS

2.1. Įrankių ir priemonių pasirinkimo analizė

Šiam darbui įvykdyti buvo pasirinkta *Bash* skriptai, nes tai buvo nurodyta darbo temos pavadinime. Kalbant apie *Bash* skriptus galima įvardinti šiuos privalumus: populiarius, nemokamas, palaikomas absoliučios daugumos *Linux/Unix* operacinių sistemų. Trumpai aptarsime kitas operacinės sistemos programinio valdymo priemones.

Csh (*C shell*) – patobulinta *tcsh* versija. *C* apvalkalas numato plačią komandinę kalbą panašią į *C* programavimo kalbą. *C* apvalkalas gali būti konstruojamas įvesčiai ir išvesčiai, sąlyginėms operacijoms, failų valdymui. Pagrindinis *Csh* tikslas yra išversti komandines eilutes įrašytas terminale, pavyzdžiui, kaip kitų programų pritaikymui. *Csh* naudinga programa bendraujant su *Unix* [20, 23].

Zsh (*Z shell*) yra galingas apvalkalas, veikia kaip interaktyvus apvalkalas. Lyginant su *Bash*, *Zsh* privalumai: greitesnis, geresnis masyvų tvarkymas, pilnai pritaikoma [24].

Ksh (*Korn shell*) – yra taip pat pilna, galinga, aukšto lygio programavimo kalba. Programos rašymas, dažnai lengviau ir greičiau vyksta nei su kitomis aukšto lygio kalbomis. Nauja *Ksh* versija turi funkcionalumą iš kitų skriptų kalbų, pvz.: *Perl*, *Rexx*, *AWK*. Dėl šios ir daugelio kitų priežasčių, *ksh* yra daug geresnė skriptų kalba, nei bet kuris kitas populiarius apvalkalas [21].

Rašyti skriptus galima ir pasinaudojus programavimo kalbas, pvz.: *Perl*, *Ruby*, *Python*.

Perl yra programavimo kalba, kuri buvo sukurta manipuliavimui tekstu, o dabar naudojama sistemų administravimui, interneto svetainių, tinklo, grafinių sąsajų programavimui ir kitoms užduotims. Šią kalbą lengva naudoti. Ši kalba yra labiau praktiška (lengva naudoti, efektyvi, pilna) negu graži (maža, elegantiška, minimalistinė) [18].

Ruby kalba yra švaresnė palyginus su *Perl* ar *Python*. Ši programavimo kalba turi stiprias teorines šaknis ir lengvą sintaksę. Su *Ruby* galima parašyti scenarijų vienai problemai spręsti, bet greičiausiai taip nedarysime. Vietoj to bus lengviau parašyti programą, kurią ateityje bus lengva suprasti, palaikyti ir išplėsti [16].

Python sparčiausiai populiarėjanti programavimo kalba pasaulyje dėl didžiulio bibliotekų kiekio. Tai leidžia supaprastinti ir pagreitinti programų kūrimą, kurti universalias, pagal poreikius pritaikytas programas. Tai programavimo kalba, kuomet kodas analizuojamas programos vykdymo metu. Be didžiulio kiekio vidinių bibliotekų yra įtraukta ir daug be tipinių duomenų struktūrų. *Python* dėka apie 25% pagreitėja programų kūrimas [5, 15].

2.2 Projekto (darbo) vykdymo planas

Darbo įvykdymui pirmiausiai buvo sudarytas darbų atlikimo planas, kuris padėjo racionaliai planuoti tolesnius darbo rašymo etapus. Po plano sudarymo sekė informacijos rinkimas, jos analizavimas ir sisteminimas. Galiausiai buvo taisomos klaidos, kuriamas darbo aprašymas.

	Spalis				Lapkritis				Gruodis				Sausis				Vasaris				Kovas				Balandis			
Sudaromas darbų atlikimo planas																												
Informacijos rinkimas ir analizavimas																												
Bash skriptų analizavimas																												
Bash skriptų kūrimas																												
Klaidų taisymas																												
Darbo aprašymo rašymas																												

Lentelė 1 Darbų įvykdymo eiga

2.3. Pradinis projekto aprašymas

Pagal darbo tematiką reikėjo naudoti *Bash* skriptus, o jie skirti *Unix/Linux* operacinių sistemų valdymui, todėl pasirinkta *Linux* operacinė sistema ir *Ubuntu* distribucija. *Ubuntu* pasirinkta, kadangi *Ubuntu* yra viena populiariausių naudojamų *Linux* distribucijų ir geriausiai pritaikyta paprasto namų vartotojo asmeniniam kompiuteriui, skirta pradedančiajam vartotojui (angl. novice-friendly), daug dokumentacijos ir vartotojų įtakojimas kuriant (angl. user-contributed) [19].

Pirmiausia buvo renkamos dažniausiai naudojamos komandos, kurios buvo sugrupuotos pagal kriterijus aprašytus 1.2 skyriuje. Norint parašyti programą, reikėjo išanalizuoti *Bash* skriptų struktūrą. Išanalizavus *Bash* skriptų struktūrą ir surinkus dažniausiai naudojamas komandas, lieka paskutinis žingsnis – jas suprogramuoti.

Išrinktos komandos yra vienos iš populiariausių naudojamų komandų, kurios reikalingos norint dirbti su *Linux/Unix* distribucijom dialogo lange (terminale). Išrinktos komandos naudingesnės pradžiamoksliam *Linux/Unix* vartotojui. Išrinktos dvidešimt septynios komandos, kurios pagal komandų paskirtį išskirstytos į penkias grupes:

1. Paprasčiausios komandos,

2. Darbo su failais komandos,
3. Tinklo procesų peržiūra ir valdymas,
4. Operacinės sistemos procesų peržiūra ir valdymas,
5. Vartotojų peržiūra ir valdymas.

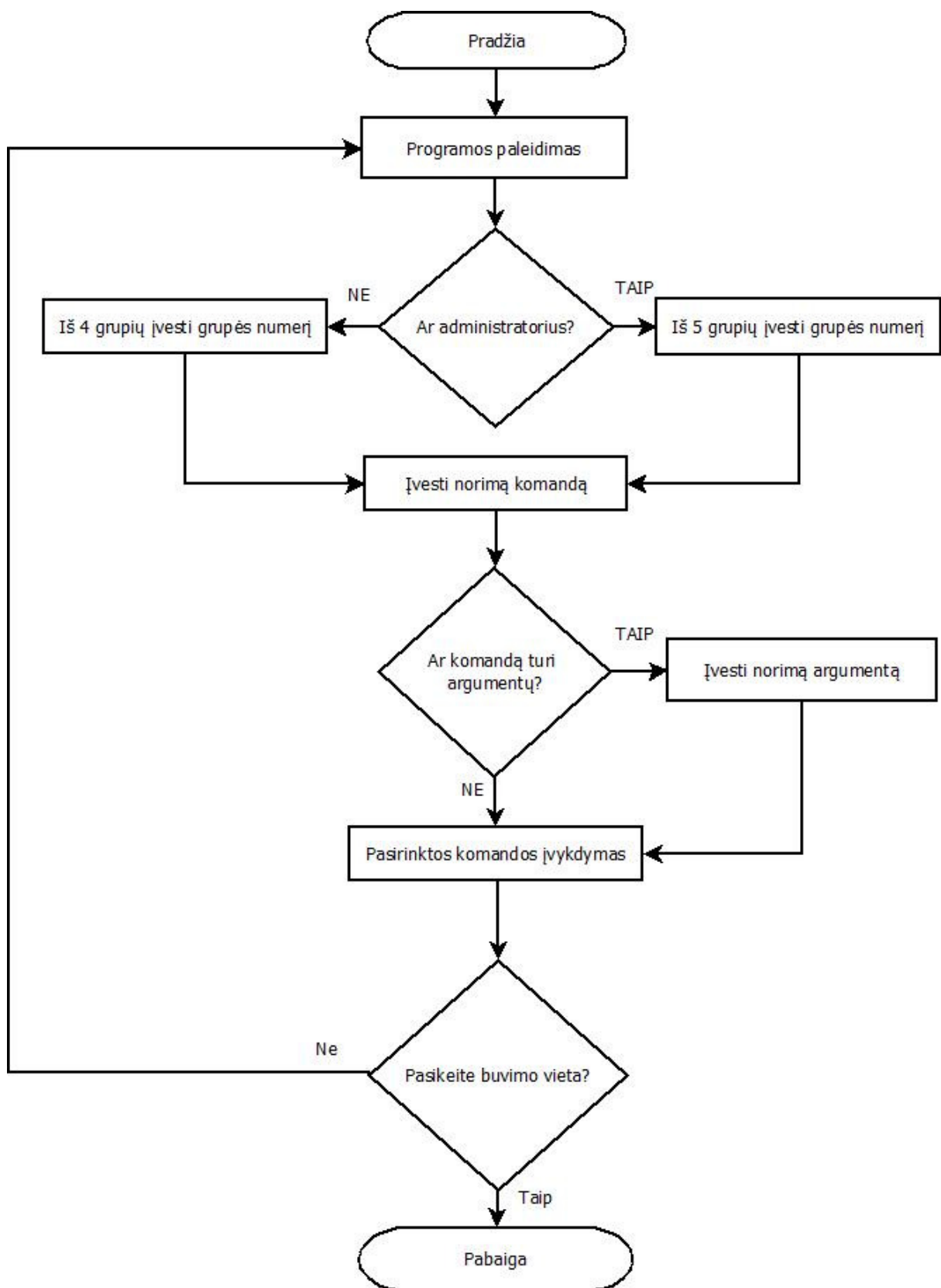
Šių grupių komandos plačiau aprašytos 2 priede.

Išanalizavus *Bash* skriptų panaudos galimybes buvo parengti keturi programinio kodo realizavimo modeliai:

- Pirmajame modelyje (toliau bus vadinama M1) vyksta komandos pasirinkimas ir rezultato gavimas;
- Antrajame modelyje (toliau bus vadinama M2) vyksta komandos pasirinkimas, parametrų nurodymas ir rezultato gavimas;
- Trečiajame modelyje (toliau bus vadinama M3) vyksta komandos pasirinkimas, parametrų nurodymas, pagal pateiktus parametrus ieškoma ir pateikiamas rezultatas;
- Ketvirtajame modelyje (toliau bus vadinama M4) vyksta komandos pasirinkimas, parametrų nurodymas, pagal pateiktus parametrus ieškoma informacija, rastos informacijos pateikimas, rinktis iš pateiktos informacijos ir išsirinktos informacijos pateikimas.

Numatyta, kad iškvietus programą patikrinama ar vartotojas yra paprastas vartotojas, ar sistemos administratorius (žr. Pav.1). Po to pateikiamas atitinkamas pasirinkimų sąrašas. Paprastam vartotojui pateikiamas komandų sąrašas, kuriam nereikia administratoriaus teisių (*Linux/Unix* sudo). Pateikus komandų grupių sąrašą galima pasirinkti grupę iš keturių ar penkių grupių arba galima išeiti. Pasirinkus grupę, pateikiamas komandų sąrašas, iš kurio reikia pasirinkti komandą. Komandos yra suprogramuotos atskirai pagal vieną iš modelių (M1, M2, M3, M4). Pasirinkus komandą yra vykdoma, jei argumentų neturi. Jei komanda turi argumentų, pateikiamas jų sąrašas, išsirinkus argumentą, tada įvykdoma komanda su pasirinktu argumentu. Įvykdžius skriptą jis bus išskviečiamas iš naujo, jei nepasikeitė vartotojo buvimo vieta, t. y., kuriame kataloge buvo, kai paleido programą, jame ir liko po komandos atlikimo (žr. Pav.1, Blokas: Pasikeitė buvimo vieta?).

Algoritmas, pagal kurį veikia programą pateiktas 1 paveiksle.

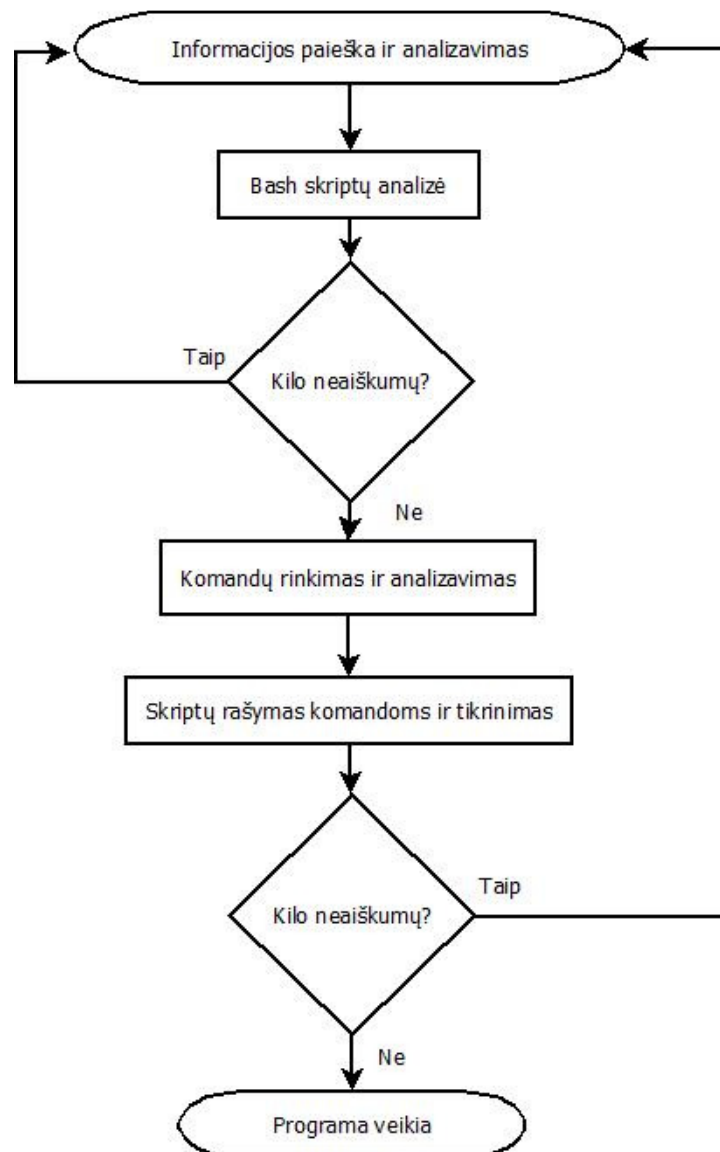


Pav. 1 Algoritmas pagal kurį veikia programa

3. DARBO EIGOS APRAŠYMAS

3.1 Darbo eigos grafas

Algoritmas nurodo, kokia tvarka buvo vykdomi darbai (žr. Pav.2). Pirmiausia buvo ieškoma medžiaga ir ji analizuojama. Toliau ši informacija buvo siejama Bash skriptų analizei. Jei kildavo neaiškumų, tekdavo grįžti prie informacijos paieškos ir analizavimo. Toliau buvo renkamos komandos, kurios bus naudojamos darbe. Atlikus šį etapą, buvo programuojamos surinktos komandos ir tikrinamos, ar teisingai veikia. Vėl kilus neaiškumams (komandų argumentų tikslumo, t. y., ką argumentas turi atvaizduoti, sintaksės klaidų, komandų nevykdymo ir kt.) buvo grįžtama prie informacijos ieškojimo. Galiausiai buvo tikrinama visa programa, ar veikia.



Pav. 2 Algoritmas kaip buvo vykdomi darbai

3.2. Problemų ir jų sprendimų aprašymai ir pagrindimai

Kadangi *Bash* skriptai nėra pritaikyti programinės įrangos kūrimui, todėl susidurta su tam tikromis problemomis, kurios būdingos tiek skriptų naudojimui, tiek *Linux* operacinės sistemos komandų valdymo sprendimams. Vienas iš minusų – lėtakai veikia, nes visos komandos interpretuojamos ir vykdomos. Iškilusios problemos rašant programą:

1. Komandai *find*, *grep* ir kt. komandoms (realizavimui naudojami M2, M3 ir M4 modeliai) reikia įvesti kelią, t. y., norint įvesti kelią, reikia visą kelią žinoti ir suvesti teisingai, todėl buvo ieškota būdų kaip palengvinti, pasufleruoti, ką reikia įvesti, kad kelias būtų nurodomas teisingai.
2. Įvykdžius *grep* ir *find* komandas jų gautus rezultatus patalpinti į masyvą, kad būtų galima pateikti vartotojui gautų rezultatų sąrašą ir jis išsirinktų, kurį failą nori atidaryti (realizavimui naudojamas M4 modelis).
3. Paleidus programą, vykdant komandas, nėra atvaizduojami gaunami rezultatai.
4. Vartotojai priskirti grupei negali naudotis grupės bendrintais katalogais.

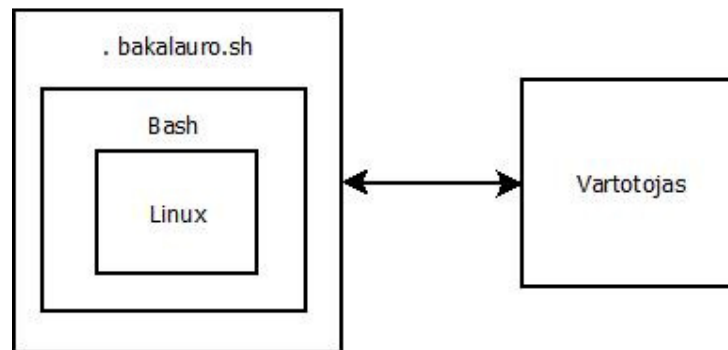
Problemų sprendimai:

1. Buvo ieškoma būdų, kaip padėti vartotojui įvesti terminale kelią. Parodyti, katalogų turinį, kas juose yra, neapsunkinant vartotojo sąsajos nereikalingais įvedimais. Buvo panaudotas *dialog* paketas. *Dialog* paketas – tai grafinės sąsajos pagalba pavaizduojami klausimai, dialogo ar pranešimų langai iš skripto apvalkalo (angl. shell). Prieš naudojant programą pirmą kartą reikia terminale įvesti komandą: *apt-get install dialog*. Jei neprisijungę kaip administratorius, tada reikia įvesti *sudo apt-get install dialog*, po to sistema paprašys įvesti administratoriaus slaptažodį. Sėkmingam problemos sprendimui reikalingas interneto ryšys.
2. Ieškant failų pagal pateiktą šabloną, ir juos pateikus, norint atidaryti reikėjo patalpinti į masyvą gautus rezultatus. Norint masyvo viduje atlikti komanda *grep*: *Grep0=(`grep -rl "\$sablonas\_grep0" \$kelias\_grep0`)*, reikia naudoti atvirkštinės kabutės (``), jos yra Bash vadinamos vykdomosiomis kabutėmis.
3. Parašyta programa rašant „bash bakalauro.sh“ ar „./ bakalauro.sh“ buvo vykdomos ne visos komandos, kurios aprašytos, pvz.: *cd*. Terminale rašant „. bakalauro.sh“ buvo įvykdomos visos aprašytos komandos, nes yra „source“ alternatyva Bash'e, kuris vykdo skriptus atidaryto bash terminalo viduje, o „bash bakalauro.sh“ ir „./ bakalauro.sh“ sukuria naują bash terminalą ir jame vykdo komandas.
4. Pasinaudojus komanda *chmod*, priskiriame bendrintiems katalogams grupę, bet grupės nariai negali prieiti prie katalogų, nes bendrintas katalogas turi grupę pagal nutylėjimą (angl.

default), t. y. grupės pavadinimas toks pat kaip vartotojo vardas. Todėl norint grupei naudotis bendrintais katalogais, reikia pasinaudoti *chgrp* komanda.

3.3. Galutinio projekto būvio aprašymas

Parašyta programa palengvina *Linux* valdymą, nes *bakalauro.sh* skriptas yra *Bash* apvalkalo komandų panaudojimo programa, naudojanti dialogo režimą. Ji padeda nepatyrusiam vartotojui paprasčiau ir lengviau valdyti *Linux* operacinės sistemos distribucijas (žr. Pav.3). *Bash* skriptų kalba neturi grafinės sąsajos ir valdymas realizuotas dialogo pagalba (terminale).



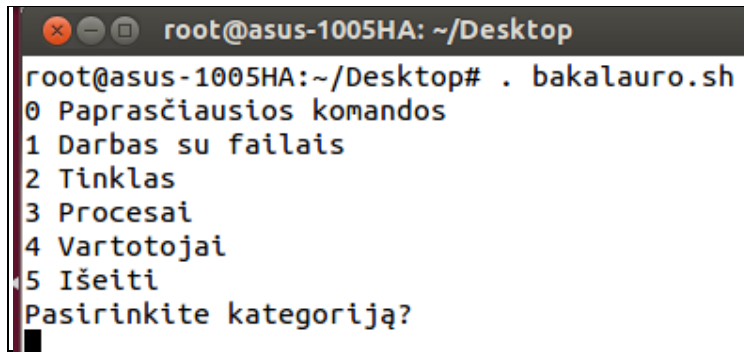
Pav. 3 Linux apvalkalų sąsaja su vartotoju

4 paveiksle pavaizduota paprasto vartotojo pasirinkimo meniu. Kaip matome, paprastam vartotojui pateikiamos keturios pasirinkimo galimybės.

```
asus@asus-1005HA: ~/Desktop
asus@asus-1005HA:~/Desktop$ . bakalauro.sh
0 Paprasčiausios komandos
1 Darbas su failais
2 Tinklas
3 Procesai
4 Išeiti
Pasirinkite kategoriją?
```

Pav. 4 Iškvietus skriptą, vartotojui pateikiamas pasirinkimo meniu

5 paveiksle pavaizduota administratoriaus pasirinkimo meniu. Administratoriui numatyta dar viena komandų grupė – *vartotojai*. Ji skirta administruoti vartotojus.

A terminal window titled 'root@asus-1005HA: ~/Desktop'. The prompt is 'root@asus-1005HA:~/Desktop#'. The user has entered '. bakalauro.sh'. The script displays a menu with the following options:

```
0 Paprasčiausios komandos
1 Darbas su failais
2 Tinklas
3 Procesai
4 Vartotojai
5 Išeiti
Pasirinkite kategoriją?
```

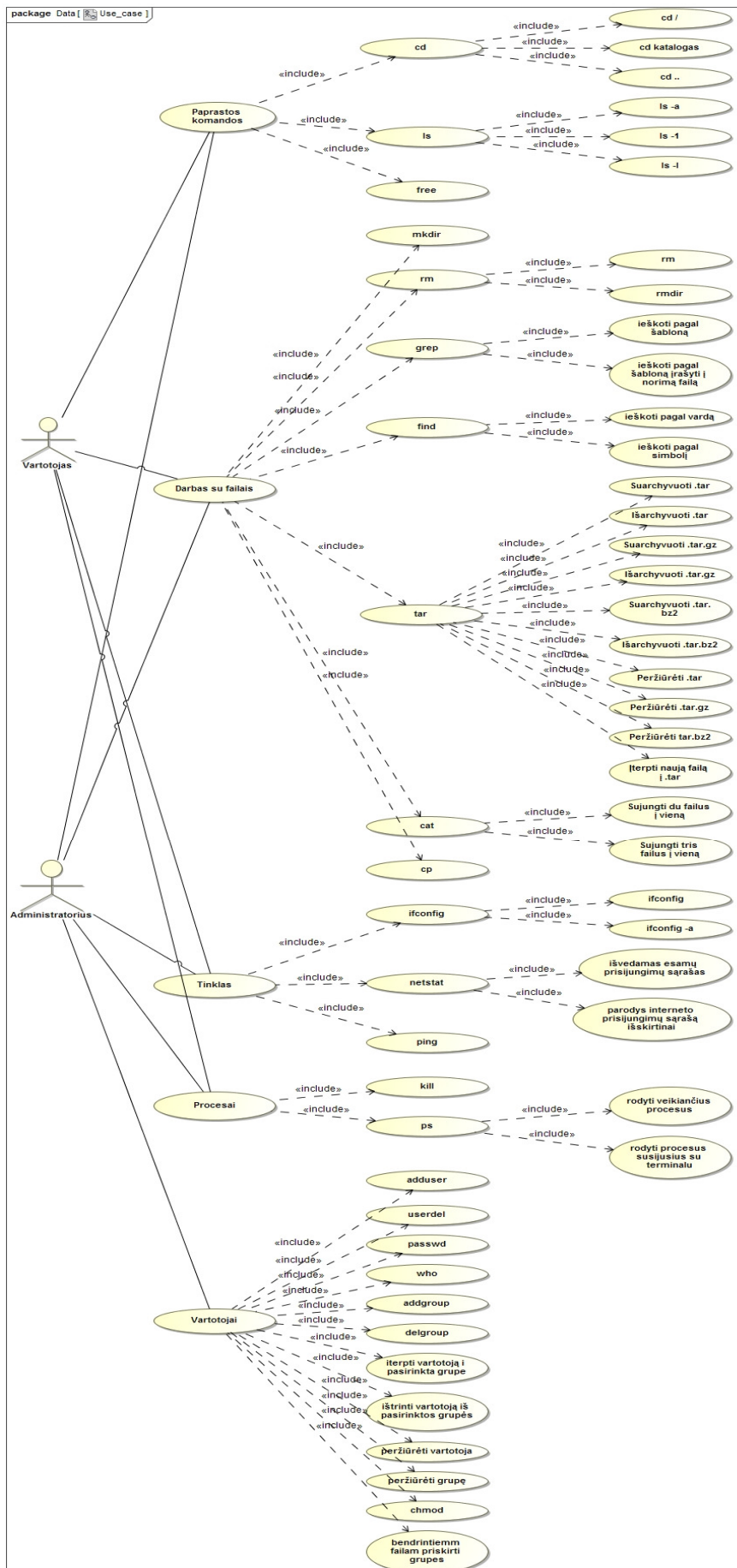
Pav. 5 Iškvietus skriptą, administratoriui pateikiamas pasirinkimo meniu

Apibendrinant galutinį projekto būvį, reikia pažymėti, kad paprasto vartotojo (žr. Pav. 4) ir administratoriaus (žr. Pav. 5) pasirinkimo meniu skirtumas yra tas, kad paprastas vartotojas nemato grupių, komandų, kurioms reikalingos administratoriaus teisės.

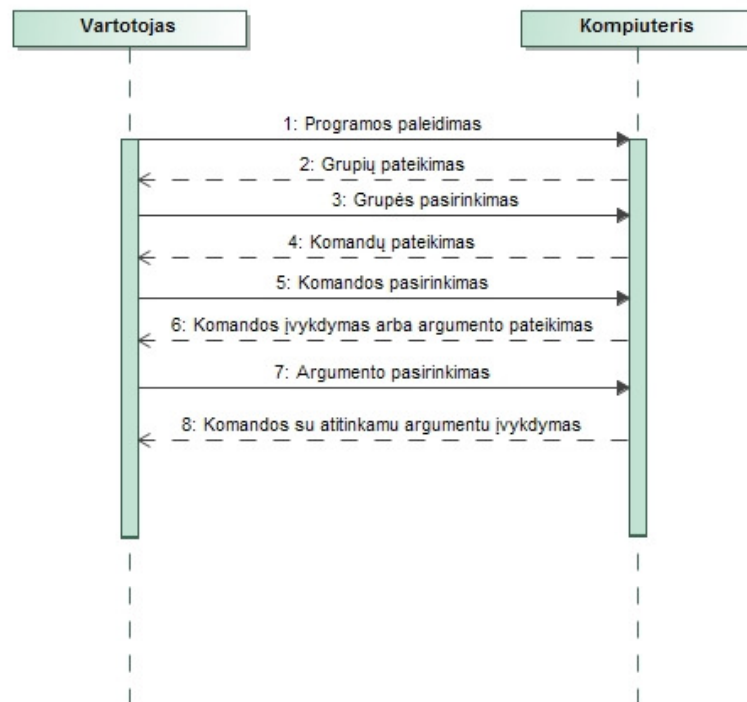
3.4. Darbo rezultatų analizė

3.4.1. Vartotojo sąsaja

Vartotojui leidžiami veiksmai pavaizduoti panaudos atvejų diagrama (užduočių diagrama) (angl. use case diagram) (žr. Pav. 6). Pavaizduota, kad pasirinkus vieną ar kitą grupę, kas iš to seka, t. y., pasirinkus vieną vartojimo atvejį, būtinai bus atliekamas ir kitas. Pvz.: vartotojui pasirinkus grupę tinklas, jam bus pateiktas komandų sąrašas: *ifconfig*, *netstat*, *ping*. Pasirinkus komandą *ping* (realizuota M2, nes komandai *ping* reikia įvesti parametrus: IP adresą arba domeno vardą (angl. domain name) ir gauname rezultatą), matome, kad argumentų komanda neturi, todėl bus įvykdyta komanda, o komandoms *netstat* (realizuota M1) ir *ifconfig* (realizuota M1, nes komandai *netstat* ir *ifconfig* nereikia nurodyti parametrų, o tik pasirenkame ir gauname rezultatą) siūloma pasirinkti argumentus. Pasirinkus argumentą, bus įvykdoma pasirinkta komanda su pasirinktu argumentu.



Programos veikimas pavaizduotas sekų diagrama (angl. sequence diagram) (žr. Pav. 7). Horizontalioji ašis rodo vaizduojamo objekto gyvavimą, tuo tarpu vertikalioji ašis rodo šių objektų sukūrimo ar iškviestos seką. Diagramoje pavaizduota programoje atsirandančių veiksmų seka. Iš sekos diagramos matyti, kad vyksta aštuoni veiksmai: pirmiausia, siunčiama užklausa kompiuteriui, t. y. programos paleidimas, kompiuteris priėmęs užklausą, nusiunčia atsakymą, pateikia grupes. Vartotojas siunčia užklausą, nurodęs grupės numerį, kompiuteris atsako, siunčia grupei priklausančias komandas. Gavus komandų sąrašą, siunčiama užklausa skaičius, kompiuteris grąžina pasirinktos komandos argumentų sąrašą arba įvykdytos komandos rezultatą. Vartotojui pasirinkus argumento skaičių vėl siunčiama užklausa ir gaunamas rezultatas, įvykdyta komanda su pasirinktu argumentu.



Pav. 7 Sekos diagrama (angl. sequence diagram), pagal kurią programa veikia

3.4.2. Testavimas

Jei vartotojas nori surasti failą ir žino jo pavadinimą, terminale paleidžia „.. bakalauro.sh“ ir pasirenka norimą kategoriją (žr. Pav. 8 a). Šiuo atveju kategorija darbas su failais skaičius 1, pasirinkus atsiranda naujas pasirinkimo langas ir renkasi, kokios jam komandos reikia, norint rasti failą, naudosis *find* pagal vardą ir spausime skaičių 3. Atsidariusiame meniu gauname du pasirinkimus:

- 0) Ieško pagal vardą, kai žinomas tikslus vardas, pvz.: testas.txt
- 1) Ieško pagal vardą, kai yra nurodomas šabloną, kuris turi būti varde, pvz.: *.txt

Vartotojas žino vardą, todėl spaudžiame skaičių ,ir lieka suvesti kelią arba palikti pagal nutylėjimą ir įvesti ieškomo failo vardą (žr. Pav. 8 b). Radus failą, jis bus atidaromas naujame lange.

```
asus@asus-1005HA: ~/Desktop
asus@asus-1005HA:~/Desktop$ . bakalauro.sh
0 Paprasčiausios komandos
1 Darbas su failais
2 Tinklas
3 Procesai
4 Išeiti
Pasirinkite kategoriją?
1
0 mkdir - sukurti failą
1 rm - trinti
2 grep - paieška
3 find - paieška
4 tar - archyvavimas
5 cat - sujungti failus
6 cp - kopijuoti failus
Pasirinkite komanda?
3
0 Ieškoti pagal vardą
1 Ieškoti pagal simbolį
Pasirinkite argumentą?
0
```

a

```
GNU nano 2.2.6 File: /home/asus/Desktop/testas
testas
Peršokti į: navigaciją, paiešką

/home/asus/

Iveskite ieškomo failo vardą
testas
0 Paprasčiausios komandos
1 Darbas su failais
2 Tinklas
3 Procesai
4 Išeiti
Pasirinkite kategoriją?
1
0 Ieškoti pagal vardą
1 Ieškoti pagal simbolį
Pasirinkite argumentą?
0

Testas (angl. test 'bandymas, mėginimas') bendrąja prasme - standartinių užduočių
Specializuotos reikšmės

Testavimas ir eksperimentavimas yra sudedamoji mokslinio metodo dalis, skirta
Ganyboje kokybės kontrolės testavimas yra procedūra, sukurta patikrinti proš
Kompiuterių moksle : kompiuterio testas - signalų rinkinys arba paprogramės
psichologijoje standartizuotos užduotys žmogaus psichinėms (IQ testas), fizš
Rinkodaroje testas reiškia tam tikrą rinkodaros tyrimo rūšį.
```

b

Pav. 8 Komandos *find* naudojimo pavyzdys

Vartotojas, žinodamas jo ieškomo failo turinį, gali ieškoti failo pagal šabloną. Pradiniai veiksmai kaip ir anksčiau minėtoje paieškoje (žr. Pav. 9 a), t. y. 1) Darbas su failais, ir pasirenkame komandą *grep*, kad ieškoti pagal šabloną failo viduje. Pasirinkus komandą *grep* gauname du pasirinkimus:

- 0) Ieškoti pagal šabloną,
- 1) Ieškoti pagal šabloną, ir visi gauti duomenys bus įrašyti į nurodytą failą.

Mes norime rasti ieškomą failą pagal šabloną, ir kad būtų pateikti duomenys ekrane, todėl spaudžiame skaičių 0. Tada suvedame kelią arba paliekame pagal nutylėjimą, ir šabloną pagal kurį ieškome (žr. Pav. 9 b). Ir visus rastus failus atspausdina sunumeruotus ekrane. Pasirenkame, kurį failą norime atidaryti.

```

asus@asus-1005HA: ~/Desktop
asus@asus-1005HA:~/Desktop$ . bakalauro.sh
0 Paprasčiausios komandos
1 Darbas su failais
2 Tinklas
3 Procesai
4 Išėiti
Pasirinkite kategoriją?
1
0 mkdir - sukurti failą
1 rm - trinti
2 grep - paieška
3 find - paieška
4 tar - archyvavimas
5 cat - sujungti failus
6 cp - kopijuoti failus
Pasirinkite komanda?
2
0 Ieškoti pagal šabloną
1 Ieškoti pagal šabloną įrašyti į norimą failą
Pasirinkite argumentą?
0

```

a

```

GNU nano 2.2.6 File: /home/asus/Desktop/test.txt
Kompiuterių moksle : kompiuterio testas - signalų rinkinys arba paprogramės, prš
psichologijoje standartizuotos užduotys žmogaus psichinėms (IQ testas), fizš
Rinkodaroje testas reiškia tam tikrą rinkodaros tyrimo rūšį.

Iveskite paleškos šablona
programos
0 /home/asus/Desktop/testa
1 /home/asus/Desktop/test.
2 /home/asus/.local/share/
3 /home/asus/.local/share/
4 /home/asus/.mozilla/fire
5 /home/asus/.mozilla/fire
6 /home/asus/.mozilla/fire
Pasirinkite kurį failą nor
1
0 Paprasčiausios komandos
1 Darbas su failais
2 Tinklas
3 Procesai
4 Išėiti
Pasirinkite kategoriją?

```

b

Pav. 9 Komandos *grep* naudojimo pavyzdys

Norint peržiūrėti procesus, reikia spausti skaičių 3, kad būtų pasirinkti kategorija, procesai (žr. Pav. 10). Ir išjungti procesą arba rodyti, kokie procesai veikia. Pasirinkus, kokie procesai veikia, galima rinktis:

- 0) Rodyti visus šiuo metu veikiančius procesus.
- 1) Rodyti visus šiuo metu veikiančius procesus, kurie susiję su terminalu.

Pasirinkus antrąją variantą, atspausdinami visi procesai, kurie susiję su terminalu. 10 paveiksle pavaizduota, kaip veikia *ps* komanda

```

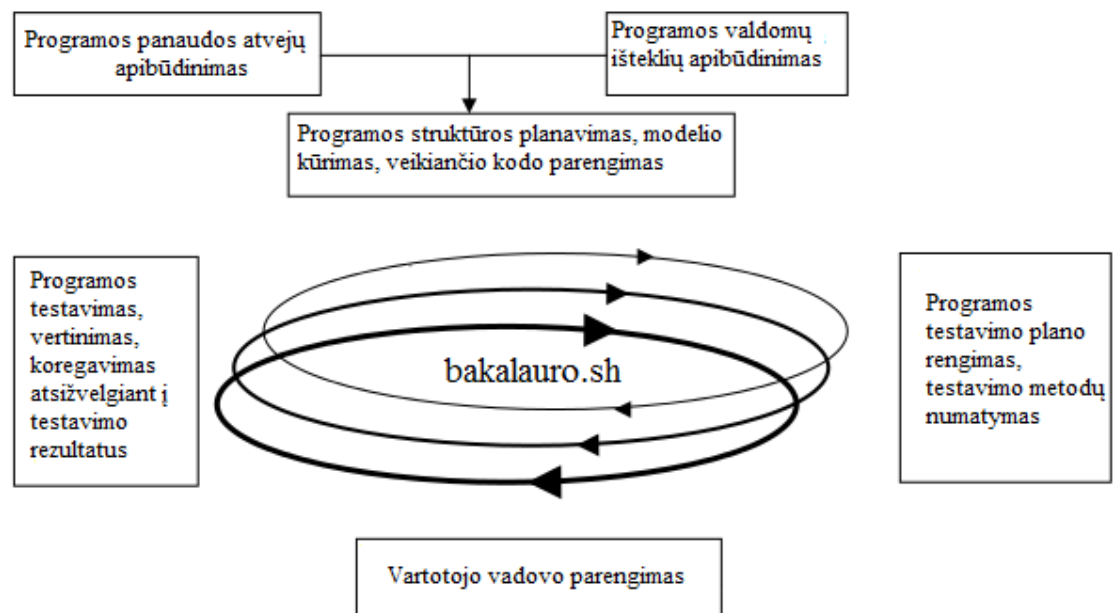
asus@asus-1005HA: ~/Desktop
asus@asus-1005HA:~/Desktop$ . bakalauro.sh
0 Paprasčiausios komandos
1 Darbas su failais
2 Tinklas
3 Procesai
4 Išėiti
Pasirinkite kategoriją?
3
0 kill - išjungti procesą
1 ps - rodyti procesus
Pasirinkite komanda?
1
0 Rodyti veikiančius procesus
1 Rodyti procesus susijusius su terminalu
Pasirinkite argumentą?
1
  PID TTY          STAT       TIME COMMAND
 3059 pts/0      Ss           0:00 bash
 3383 pts/0      R+           0:00 \_ ps Tf
 3384 pts/0      D+           0:00 \_ [more]
0 Paprasčiausios komandos
1 Darbas su failais
2 Tinklas
3 Procesai
4 Išėiti
Pasirinkite kategoriją?

```

Pav. 10 Pavyzdys, kaip veikia *ps* komanda

3.5. Programos naudojimo sritis

Programa skirta pradedančiajam administratoriui mokymosi tikslais. Padeda valdyti *Unix/Linux* operacinę sistemą, suvokti operacinių sistemų ypatumus. Taip pat įgauti naujų žinių ir lavinti įgūdžius, juos pritaikant praktiškai. Rengiant programą buvo remiamasi Wagennar darbo kokybės užtikrinimo modeliu, kuris padėjo planuoti programos struktūrą, kuri būtų lengviausiai suvokiama vartotojui, atsižvelgiant į *Bash* skriptų galimybes. Šis modelis remiasi „Tuning“ studijų programos kokybės nuolatinio tobulinimo ypatumais, kurie skatina gerinti programos kokybę. Programą testavau ne tik aš, bet ir kiti kompetentingi asmenys, kurie išsakė savo nuomonę ir pastabas, kurios pravertė koreguojant patį programos kodą. Šis modelis pagrįstas prielaida, jog programa gali ir turėtų būti tobulinama, remiantis ne tik kolegų rezultatais, bet ir tikėtinais pokyčiais. Grįžtamojo ryšio tikslas yra nustatyti ir pašalinti programos struktūros trūkumus, o prognozuojamas ryšys yra skirtas tikėtiniems pokyčiams, į kuriuos vertėtų atsižvelgti tobulinant programą. Buvo parengtas ir vartotojo vadovas, kuris leis pasinaudoti sukurta programa ir gana silpnus darbo su *Linux/Unix* įgūdžius turintiems vartotojams [13].



Pav. 11 Darbo kokybės užtikrinimo modelis (Wagenaar, 2008)

3.6. Patarimai, pastebėjimai, rekomendacijos

Svarbu atkreipti dėmesį į *Bash* sintaksę, kuri yra griežta kaip ir daugelio kitų programavimo kalbų. Pavyzdžiui:

```
if [ $kategorija -le $ELEMENTS ]
```

Kaip matoma pavyzdyje rašant komandą būtina dėti laužtinius skliaustus ir viduje skliaustų dėti tarpus. Operandas „-le“ reiškia mažiau arba lygu „<=“ taikomi aritmetinių veiksmų palyginimui.

Ilgas komandų sąrašas anglų kalba. *Bash* skriptų programavimui galima panaudoti didelį kiekį komandų tiek pagrindinių, tiek papildomų.

Galima skriptą iškviesti naudojant „bash vardas.sh“, „./ vardas.sh“ arba „. vardas.sh“, bet geriausia iškviesti skriptą „. vardas.sh“, nes *Bash* turi „source“ alternatyva, kuris vykdo skriptus atidaryto *Bash* terminalo viduje, o „bash vardas.sh“ ir „./ vardas.sh“ sukuria naują *Bash* terminalą ir jame vykdo komandas.

IŠVADOS

1. Išanalizavus teorinę medžiagą, išsiaiškinus *Bash* skriptų struktūrą, išrinktos dažniausiai naudojamos komandos, jos suskirstytos į grupes.
2. Suprojektuotas *Linux/Unix* operacinės sistemos valdymo modelis, parengtas veikiantis kodas, atsižvelgiant į išskirtas komandų grupes bei *Bash* skriptų naudojimo galimybes.
3. Išanalizavus šiuo metu siūlomas *Unix/Linux* distribucijas, pasirinkta *Ubuntu*, nes populiariausia iš visų distribucijų, geriausiai pritaikyta paprasto namų vartotojo asmeniniam kompiuteriui.
4. Programa ištestuota, vykdant testavimą, išspręstos keturios problemos, kurios vykdymo metu pateikdavo neteisingus rezultatus.

Literatūros ir informacinių šaltinių sąrašas

Literatūra

1. Cooper M., *Advanced Bash-Scripting Guide*, 2002. Prieiga internetu <<http://www.tldp.org/LDP/abs/abs-guide.pdf>> [žiūrėta 2013-04-15].
2. Garrels M., *Bash Guide for Beginners*, 2008. Prieiga internetu <<http://www.tldp.org/LDP/Bash-Beginners-Guide/html/Bash-Beginners-Guide.html>> [žiūrėta 2013-04-15].
3. Newham C., Rosenblatt B., *Learning the bash Shell, 3rd Edition*, O'REILLY, 2005.
4. Robbins A., *Bash Quick Reference*, O'REILLY, 2006.

Informacijos šaltiniai

5. *About Python*, Prieiga per internetą <<http://www.python.org/about/>> [Žiūrėta 2013-04-22]
6. *An A-Z Index of the Bash command line for Linux*. Prieiga per internetą <<http://ss64.com/bash/>> [Žiūrėta 2013-04-23]
7. Ashley J.S Mills, *Unix Shell Scripting Tutorial*, 2005. Prieiga per internetą <<https://supportweb.cs.bham.ac.uk/documentation/tutorials/docsystem/build/tutorials/unixscripting/unixscripting.html>> [Žiūrėta 2013-04-23]
8. *Dažniausiai naudojamos Linux komandos*. Prieiga per internetą <http://www.akmc.lt/akmc_wordpress/turinys/uploads/2013/02/komandos.pdf> [Žiūrėta 2013-04-23]
9. Kriauciūnas M., *Pagrindinės Linux komandos, darbas tekstiniame lange* Prieiga per internetą <http://ftp.akl.lt/Dokumentacija/Linux-serverio-seminarai/pagrindines_linux_komandos.pdf> [Žiūrėta 2013-04-23]
10. Labas.com, *Kas yra tas Linux?*, 2003. Prieiga per internetą <http://www.straipsniai.lt/Operacines_sistemas/informacija/linux_unix/puslapis/2036> [Žiūrėta 2013-04-22]
11. *Linux Distributions*. Prieiga per internetą <http://linuxlookup.com/linux_distributions> [Žiūrėta 2013-05-23]
12. Lutus P., *Bash Shell Programming in Linux*, 2006. Prieiga per internetą <http://arachnoid.com/linux/shell_programming.html> [Žiūrėta 2013-04-23]
13. Milišiūnaitė I., Butkienė J., Juknytė-Petreikienė I., Keturakis V., Lepaitė D., *Kompetencijų plėtotės ir studijų siekinių vertinimo metodikos integravimo į vidinio kokybės užtikrinimo sistemą rekomendacijos*, 2010. Prieiga per internetą <<http://www4066.vu.lt/Files/File/Metodikos%20integravimas.pdf>> [Žiūrėta 2013-05-20]

14. Noreika A., *Linux distribucijos – kas tai?*, 2010. Prieiga per internetą <<http://www.technologijos.lt/n/technologijos/it/S-13663>> [Žiūrėta 2013-41-22]
15. Pralgauskis J., *Python – programavimas visiems*, 2008. Prieiga per internetą <http://www.technologijos.lt/n/technologijos/it/straipsnis/Python--programavimas-visiems?name=straipsnis-2447&l=2&utm_source=Susije_po_straipsniu&utm_medium=Vidine_navigacija&utm_campaign=Vidine_navigacija> [Žiūrėta 2013-04-15]
16. *Programming Ruby. Preface*. Prieiga per internetą <http://www.ruby-doc.org/docs/ProgrammingRuby/> > [Žiūrėta 2013-04-23]
17. Rouse M., *Script*, 2005. Prieiga per internetą <<http://searchenterprise.linux.techtarget.com/definition/script>> [Žiūrėta 2013-04-22]
18. Sheppard D., *Beginner's Introduction to Perl*, 2000. Prieiga per internetą <<http://www.perl.com/pub/2000/10/begperl1.html>> [Žiūrėta 2013-04-23]
19. *Top ten distributions*. Prieiga per internetą <<http://distrowatch.com/dwres.php?resource=major>> [Žiūrėta 2013-04-23]
20. *Unix: C-Shell Scripts*. Prieiga per internetą <http://www.cims.nyu.edu/~ytang/c_script.html> [Žiūrėta 2013-04-22]
21. *What is the KornShell Language?*. Prieiga per internetą <<http://www.kornshell.com/info/>> [Žiūrėta 2013-04-22]
22. *What is Unix*, 2012. Prieiga per internetą <<http://kb.iu.edu/data/agat.html>> [Žiūrėta 2013-04-23]
23. William Joy, *An Introduction to the C shell*. Prieiga per internetą <<http://www.kitebird.com/csh-tcsh-book/csh-intro.pdf>> [Žiūrėta 2013-04-23]
24. *Zsh*. Prieiga per internetą <<https://wiki.archlinux.org/index.php/Zsh>> [Žiūrėta 2013-04-23]

Anotacija (Summary)

Autorius: Mindaugas Gapšys

Tema: Bash skriptų panaudojimas Unix/Linux operacinių sistemų valdymui

Šiaulių universitetas 2013 metai

Darbo tikslas – naudojant Bash skriptus parengti Linux/Unix operacinių sistemų valdymo programą. Buvo analizuota Bash skriptų struktūra, parengti operacinės sistemos valdymo modeliai, pagal kuriuos parašyta programa.

Author: Mindaugas Gapšys

Topic: The usage of Bash scripts for Unix/Linux the operating systems management

Šiauliai university 2013 year

The aim of the work is the creation of the Unix / Linux operating systems management program by using Bash scripts. The structure of Bash scripts was analysed as well as the models of the operating system management according to which the program was written were made.

Priedai

1 priedas. CD Turinys

- Aprašas – kataloge yra darbo aprašymas (docx ir pdf formatais).
- Programa – kataloge yra parašyta programa „bakalauro.sh“ *Linux/Unix* operacinei sistemai valdyti.
 - VirtualBox – kataloge yra *Linux Ubuntu* distribucijos suarchyvuota virtuali mašina „Ubuntu_su_pagalba.rar“. Leidžiant reikia nustatyti atminties mažiausiai 1024 MB, paleidus mašina reikia turėti kantrybės, nes lėtai veikia mašina. Pasileidus *Linux* atsiradome terminalo langą ir suvedame *cd Desktop*, kad pakilti lygmeniu ir paleidžiame programą . *bakalauro.sh*. Administratoriaus slaptažodis: *adminuser*.
- Turinys.txt – tekstinis failas, kuriame yra CD turinys

2 priedas. Vartotojo vadovas

Atsidariusiame terminalo lange suvedame . *bakalauro.sh*, tada bus pradedama vykdyti programa ir pateiks grupių sąrašą, atitinkamai kokios prisijungusio vartotojo privilegijos (paprastas vartotojas / administratorius). Pasirinkus grupės numerį, bus pateiktos komandos. Išsirinkta komanda bus įvykdyta ir atvaizduoti gauti rezultatai, arba bus pateikiami komandos argumentai ir išsirinkus argumentą, bus pradedama vykdyti komanda su pasirinktu argumentu. Norint įvykdyti komandą reikia įvesti papildomus parametrus, pvz.: katalogo kelią, failo vardą ir kt. Įvykdžius komandą automatiškai iškviečiama programa iš naujo, jei nepasikeitė buvimo vieta. Komandos, kurios sudaro grupes, skliausteliuose nurodyta, pagal kokį modelį suprogramuota:

0) Paprastos komandos

0) cd – perėjimas

- 0) cd / – Pakilti į patį aukščiausią šakninį katalogą (M1).
- 1) cd katalogas – įeina į nurodytą katalogą (M2).
- 2) cd .. – nuleidžia vienu lygmeniu žemyn (M1).

1) ls – rodyti turinį

- 0) ls-a – parodo visus, įskaitant paslėptuosius failus (M1).
- 1) ls -l – rodo failus viename stulpelyje (M1).
- 2) ls-l – informacija apie failus kataloge (M2).

2) free – operatyvinės atminties būklės (RAM) panaudojimas megabitais (M1).

1) Darbas su failais

0) mkdir – sukurti failą (M2).

1) rm - trinti

- 0) rm – trinti failą (M2).
- 1) rmdir– trinti katalogą (M2).

2) grep – paieška

- 0) ieškoti pagal šabloną (M4).
- 1) ieškoti pagal šabloną ir įrašyti į norimą failą (M3).

3) find – paieška

- 0) ieškoti pagal vardą (M3).
- 1) ieškoti pagal šabloną (M4).

4) tar - archyvavimas

- 0) suarchyvuoti .tar (M3).
- 1) išarchyvuoti .tar (M3).
- 2) suarchyvuoti .tar.gz (M3).
- 3) išarchyvuoti .tar.gz (M3).
- 4) suarchyvuoti .tar.bz2 (M3).
- 5) išarchyvuoti .tar.bz2 (M3).
- 6) peržiūrėti .tar (M2).
- 7) peržiūrėti .tar.gz (M2).
- 8) peržiūrėti .tar.bz2 (M2).
- 9) įterpti naują failą į .tar (M3).

5) cat – sujungti failus

- 0) sujungti du failus į vieną (M2).
- 1) sujungti tris failus į vieną (M2).

6) cp – kopijuoti failus (M2).

2) Tinklas

0) ifconfig – naudojama tinklo nustatymų peržiūrėjimui

- 0) ifconfig – naudojama tinklo nustatymų peržiūrėjimui bei konfigūravimui (M1).
- 1) ifconfig-a – rodo visus įrangos nustatymus (M1).

1) netstat – išvedamas esamų prisijungimų sąrašas

- 0) išvedamas esamų prisijungimų sąrašas (M1).
- 1) parodo interneto prisijungimų sąrašą išskirtinai (M1).

2) ping – komanda tikrinti tinklo ryšius (M2).

3) Procesai

0) kill– išjungti procesą (M2).

1) ps – rodyti procesus (M1).

- 0) rodyti veikiančius procesus (M1).
- 1) rodyti procesus, susijusius su terminalu (M1).

4) Vartotojai

- 0) adduser – sukurti vartotoją (M2).
- 1) userdel – panaikinti vartotoją (M2).
- 2) passwd – keisti slaptažodį (M2).
- 3) who – parodo, kas yra prisijungęs prie sistemos (M2).
- 4) addgroup – sukurti naują grupę (M2).
- 5) delgroup – panaikinti grupę (M2).
- 6) įterpti vartotoją į grupę (M2).
- 7) ištrinti vartotoją iš grupės (M2).
- 8) peržiūrėti vartotoją (M3).
- 9) peržiūrėti grupę (M3).
- 10) chmod – keisti teises (M2).
- 11) bendrintiems failams priskirti grupės (M2).