

4. OS LINUX komandos

4.1. Operacinių sistemų komandinė kalba

Su Linux klasės sistemomis leidžiama dirbti tik šioje sistemoje užregistruotiems vartotojams. Paprastai vartotojus registruoja sistemos administratorius, kuris registruojamam vartotojui suteikia prisijungimo *login* vardą (6 - 8 simboliai) bei slaptažodį (6 - 8 simboliai). Slaptažodis turi užtikrinti vartotojo autentiškumą, todėl jis parenkamas laikantis tam tikrų saugos reikalavimų: jis sudaromas naudojant didžiąsias, mažąsias raides, skaičius, specialius simbolius. Įvedęs prisijungimo vardą bei slaptažodį (kuris ekrane įvedimo metu gali būti niekaip neindikuojamas), vartotojas patenka į jam administratoriaus priskirtą namų katalogą, kuriame gali kurti savus žemesnio lygio katalogus bei juose laikyti savus failus. Kai kurie vartotojo darbui su sistema reikalingi failai jau būna jo namų kataloge.

Prisijungęs prie sistemos, vartotojas įgyja visas jam administratoriaus suteiktas privilegijas. Vartotojai LINUX sistemoje yra skirstomi į grupes, taigi kiekvienas vartotojas, be savo privilegijų dar turi ir tas privilegijas, kurios yra priskirtos jo grupei, bei privilegijas, kurios priklauso visiems sistemos vartotojams.

Baigęs darbą su sistema, vartotojas turi atsijungti. Atsijungti nuo sistemos galima komanda **logout (exit)**.

Prisijungus prie sistemos, reikia atlikti tam tikrus veiksmus. Taigi trumpai aptarkime Linux komandas. Su operacine sistema galima bendrauti komandine kalba – įvedant komandas. Komandinė kalba leidžia vartotojui „paprasyti“ kompiuterį įvykdyti tam tikrus veiksmus. Kompiuteris taip pat naudoja tam tikrą kalbą atsakyti į šias komandas. Dažniausiai šias abi kalbas valdo *Komandų interpretatorius*. Linux sistemoje komandų interpretatoriaus vaidmenį atlieka **shell** programa (apvalkalas).

Komandinė kalba priima simbolių eilutes. Tikimasi, kad pirmasis įvestas eilutės žodis atitiks vieną iš operacinės sistemos naudojamų komandos vardų. Bendra LINUX komandos forma yra ši:

komanda [parinktys] [parametrai].

Parinktys nurodo komandos režimą; jos paprastai rašomos po minuso ženklo ir žymimos raidėmis.

Nors dėl istoriškai susiklosčiusių aplinkybių yra išimčių, tačiau beveik visos komandos Linux sistemoje vartojamos pagal tam tikras taisykles:

- komandos ilgis gali svyruoti nuo 2 iki 9 simbolių;
- komanda rašoma tik mažosiomis raidėmis;
- parinktys gali susidėti tik iš vienos raidės, prieš kurią yra minuso ženklas;
- po vieno „-“ ženklo gali būti nurodytos kelios parinktys, kurių eiliškumas neturi reikšmės. Tarp kelių nurodomų parinkčių tarpai nepaliekami. Pavyzdžiui, komandoje:

ls -al /dev

ls yra komanda, liepianti parodyti katalogo **/dev** turinį, parinktys „**a**“ liepia rodyti ir paslėptus failus, parinktys „**l**“ liepia informaciją pateikti plačiu išvedimo formatu.

Linux sistema yra jautri registrai, todėl kai vietoj komandos „ls“ įvedame „LS“, tai ji interpretuojama kaip bloga komanda. Failų vardai „**pirmas**“ ir „**Pirmas**“ Linux sistemoje taip pat suprantami kaip vardai, žymintys du skirtingus failus.

Linux vartotojai gali naudotis sistemoje esančiu virtualiu vadovėliu, kuriame galima rasti informaciją apie pagrindines Linux komandas, funkcijas, kreipinius ir kt. Informacija iš vadovėlio pateikiama, naudojant komandą **man**. Ieškoma informacija apie dominantį objektą pateikiama ekrane teksto pavidalu.

Komanda **man** galima naudotis taip:

man topic

Čia vietoj žodžio **topic** reikia įrašyti žodį (komandos vardą, funkcijos vardą ar kt.), apie kurį ieškosite informacijos vadovėlyje. Pavyzdžiui, norint rasti visą informaciją apie komandą **ls**, reikia įvesti šią komandą:

man ls

Vadovėlyje galima paieška ir pagal raktinį žodį. Tokia paieška galima pasinaudoti tada, kai nesate tikri, ar iš tikrųjų tiksliai žinote komandos vardą. Šiuo atveju bus pateiktas komandų, susijusių savo vardu arba aprašymu su šiuo raktiniu žodžiu, sąrašas, nurodomas trumpas šių komandų aprašymas bei skyrius, kuriame reiktų ieškoti detalesnės informacijos:

man -k raktinis_žodis | more

Šioje komandoje esanti komanda **more** išveda informaciją pateiks ekrane puslapiais. Norint skaityti kitą puslapį, reikės spausti tarpo klavišą. Vietoj šios komandos paieškai pagal raktinį žodį galima naudoti ir komandą **apropos**:

apropos raktinis_žodis | more

Abiem šiais atvejais reikia įrašyti tą raktinį žodį, pagal kurį norite atlikti paiešką.

Informacija iš vadovėlio yra išvedama puslapiais. Norint tęsti jos išvedimą, reikia spausti tarpo klavišą, norint skaityti kitą eilutę, reikia spausti „**Enter**“ klavišą, o, norint nutraukti išvedimą reikia spausti „**Ctrl+C**“ klavišus.

Norint pasiskaityti, kaip naudotis vadovėliu, reikia įvesti komandą:

man man

Vadovėlyje esanti informacija yra suskirstyta į skyrius. Jei kreipdamiesi į vadovėlį nenurodote skyriaus, tai bus pateikta pirmame pasitaikiusiame skyriuje surasta tinkama informacija. Pavyzdžiui, Linux turi komandą **write** bei funkciją **write**. Todėl, norint gauti informaciją apie funkciją **write**, reikia nurodyti, kad reikalinga informacija iš atitinkamo skyriaus.

Norint gauti informaciją apie objektą iš atitinkamo skyriaus, galima naudoti **man** komandą su parinktimi **-s**:

man -s5 ms

man -s3SOCKET socket

Pirmojo skyriaus nurodyti nereikia, nes informacijos paieška ir pradedama nuo pirmojo skyriaus.

Aprašant komandas, gali būti panaudoti laužtiniai skliaustai – jie reiškia, kad atitinkama komandos dalis nėra būtina, pavyzdžiui:

rm [-fri] file ...

Tai rodo, kad komanda **rm** gali turėti tris parinktis (**f**, **r**, ir **i**), kurias galime naudoti arba nenaudoti. Tai yra bus teisingos visos šios komandos:

rm xyz

rm test1 test2 test3

```
rm -f m.test
rm -ri lab_d csort.c f1
```

4.2. Pagrindinės komandos

Komandų įvairovė yra gana didelė. Pateiksime kai kurias dažnai naudojamas komandas ir trumpą jų aprašymą.

4.2.1. Komandos veiksams su katalogais

Aptarsime komandas, skirtas dirbti su katalogais. Tai šios komandos:

pwd, ls, cd, mkdir, rmdir	
mkdir	sukuria naują katalogą;
rmdir	ištrina katalogą
pwd	atspausdina einamąjį katalogą;
ls	parodo katalogo turinį;
cd	pereina į nurodytą katalogą.

Jei dar nieko neturite savame kataloge, tai pabandykite patikrinti, kurioje failų sistemos vietoje yra jūsų namų katalogas – įveskite komandą

```
pwd
```

Sistema atsilieps nurodydama jūsų namų katalogo vietą, pavyzdžiui:

```
/home/kostas
```

Ši informacija rodo, kad šis namų katalogas **kostas** yra pakatalogis **home** kataloge, o šis yra katalogų medžio viršuje, vadinamajame šaknies kataloge (/). Apie patį failų medį dar pakalbėsime šiek tiek vėliau.

Komanda **ls** (nuo žodžio *list*), panaudota be parinkčių, iš pradžių tikriausiai nieko neišves, jei savo namų kataloge dar nespėjote sukurti failų. Norėdami pamatyti vadinamuosius paslėptus failus, kurių vardai prasideda taško simboliu, galite surinkti komandą

```
ls -a
```

4.3.1.5. Komanda mkdir

Komanda **mkdir** skirta naujiems katalogams kurti. Pavyzdžiui:

```
mkdir naujas
```

Bus sukurtas naujas katalogas einamajame kataloge vardu **naujas**. Įsitikinti galima panaudojus komandą **ls**. Komanda leidžia kurti iš karto kelis naujus katalogus. Pavyzdžiui:

```
mkdir lab1 lab2
```

```
ls
```

Bus sukurti du katalogai **lab1** ir **lab2**. Komanda **ls** rodys, kad šie katalogai atsirado jūsų einamajame kataloge.

4.3.1.6. Komanda rmdir

Ji naudojama katalogui šalinti, tačiau ji gali pašalinti tik tuščią katalogą. Pavyzdžiui, norime pašalinti katalogą **lab2**, esantį einamajame kataloge. Veiksmų seka gali būti tokia:

```
cd lab2
pwd
rm *
cd ..
rmdir lab2
```

Iš pradžių komanda **rm** išvalome kataloge **lab2** esančius failus, paskui „pakylame“ į aukštesnį katalogą ir jame ištriname katalogą **lab2**.

4.3.1.7. Komanda ls

Prisijungus prie sistemos, patenkama į savo namų katalogą. Šio katalogo vardas dažniausiai sutampa su vartotojo prisijungimo vardu. Jo vardą bei vietą failų sistemoje nustato sistemos administratorius, užregistrudamas jus kaip sistemos vartotoją. Namų katalogas yra skiriamas vartotojo reikmėms, jame vartotojas gali saugoti savo failus, kurti žemesnio lygio pakatalogius.

Norint pažiūrėti, kas yra šiame kataloge, galima pasinaudoti komanda **ls**.

Ši komanda išves jūsų namų katalogo turinį. Jei kataloge nebus failų (gali būti tik paslėpti failai), tai ši komanda nieko neišves.

Paprastai sistemos administratorius į naujo vartotojo namų katalogą įkelia kai kuriuos apvalkalui reikalingus paslėptus failus. Norint peržiūrėti jų sąrašą, reikalinga komanda

ls -a

Ši komanda parodo visus failus, kurių vardai prasideda tašku (paslėptus failus).

Komandos **ls** išvedimas yra susijęs su panaudotomis parinktimis. Kai kurios parinktys:

Parinktis	Reikšmė
-l	Katalogo turinys išvedamas vienu stulpeliu
-a	Išvedami visi failai, taip pat ir paslėptieji
-l	Išveda ilgu formatu. Informacija yra tiek apie failus, tiek apie pakatalogius
-r	Pateikia failus atvirkščio išdėstymo tvarka
-R	Rekursyvus išvedimas, rodomi pakatalogių turiniai
-s	Rodo failų dydį blokais

4.3.1.8. Komanda cd

Komanda **cd** (*change directory*) leidžia pereiti į kitą katalogą. Galimi tokie šios komandos naudojimo variantai:

cd katalogas

Šiuo atveju bus pereinama į nurodytą katalogą, kuris turi būti einamojo katalogo pakatologis. Katalogas gali būti nurodomas visu keliu, pavyzdžiui:

```
cd /dev/pseudo
```

arba vien katalogo vardu, pavyzdžiui:

```
cd lab1
```

Šiuo atveju bus suprantama, kad **lab1** - tai katalogas, esantis einamajame kataloge.

Komanda **cd** gali būti naudojama ir taip:

```
cd ..
```

```
cd .
```

```
cd
```

```
cd ~vart
```

Du taškai komandoje **cd** žymi vienu lygiu aukštesnį katalogą, esantį failų sistemoje, vienas taškas žymi einamąjį katalogą. Komanda **cd** be argumentų leidžia vartotojui grįžti į savo namų katalogą, **~vart** žymi vartotojo **vart** namų katalogą.

4.3. *Tekstų redaktorius*

Tekstų redaktorius **pico**.

Pateiksime šiek tiek informacijos apie šį redaktorių, kuri padės greitai suvokti, kaip šį redaktorių panaudoti. Detalesnę informaciją galima gauti įvedus komandą

```
man pico
```

Pradėti redaguoti naują failą galima įvedus komandą

```
pico naujas_failas
```

Šiuo atveju bus kuriamas naujas failas vardu **naujas_failas**.

Redaktoriaus aplinka yra labai paprasta: tekstas įvedamas klaviatūra, trinamas *backspace* arba *delete* klavišais, judama krypties klavišais, į naują eilutę pereinama paspaudus *Enter*. 1 pav. parodytas **pico** redaktoriaus langas. Redaktoriaus komandos matomos ekrano apačioje – jos apibūdinamos klavišu Ctrl (rodoma simboliu ^), naudojamu kartu su raide. Pavyzdžiui, norint išsaugoti, reikės komandos Ctrl+O (tai veiksmas *Save*). Ctrl+X naudojama darbui baigti su **pico** redaktoriumi. Jei prieš tai neišsaugojote pataisymų, bus paklausta, ar tai padaryti. Jei nebuvote nurodę failo vardo, kurdami naują failą, būsite paklaustas, koku vardu išsaugoti failą.

```
UW PICO(tm) 4.8                               File: naujas_failas

[ New file ]
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Pg  ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify    ^W Where is  ^V Next Pg   ^U UnCut Text^T To Spell
```

1 pav. **pico** redaktoriaus darbo langas

Norint redaguoti jau esamą failą, rašoma:

pico egzistuojantis_failas

Pabandykite sukurti failą vardu **tekstas**. Į jį įrašykite kelis žodžius naudodami **pico** redaktorių.

4.3.1. Veiksmų su failais komandos

cat	išveda failo turinį;
more	failo turinį išveda puslapiais;
head	išveda pirmas failo eilutes;
tail	išveda paskutines failo eilutes;
file	parodo failo tipą;
chown	keičia failo savininką;
chgrp	keičia failo grupės identifikatorių;
rm	išmeta failą;
cp	kopijuoja failą;
mv	perkelia failą;
chmod	pakeičia failo prieinamumą;
touch	keičia failo modifikavimo laiką.

Pavyzdžiai:

```
ls
aa lab pranesimai
file aa
aa:                commands text
file lab
lab:               directory
file pranesimai
```

```
pranesimai:      ascii text
cp pranesimai lab/
cp /etc/passwd lab/
```

Čia komanda **ls** rodė, kad einamajame kataloge yra failai **aa**, **lab** bei **pranesimai**. Komanda **file** rodo, kokio tipo yra šie failai. Komanda **cp** kopijuoja failą **pranesimai** į pakatalogį **lab**. Kita **cp** komanda kopijuoja failą **passwd**, esantį kataloge **/etc** į pakatalogį **lab**.

4.3.1.5. Failo peržiūros komanda cat

Komanda skirta mažiems failams peržiūrėti. Norint išvesti failo **tekstas** turinį ekrane, rašoma:

```
cat tekstas
```

Šią komandą galima naudoti ir keliems failams sujungti, pavyzdžiui, norint sujungti failus **tekstas1** ir **tekstas2** į vieną failą **tekstas**, rašoma:

```
cat tekstas1 tekstas2 > tekstas
```

Šioje komandoje panaudotas ženklas **>** rodo, kad rezultatai, užuot išvesti į ekraną, yra nukreipiami į failą **tekstas**.

Galima šią komandą naudoti ir trumpiems failams sukurti. Pavyzdžiui:

```
cat >tekstas3
```

```
alfa
```

```
beta
```

```
Ctrl+D
```

Šiuo atveju pirmąją komandos eilutę nurodoma, kad iš klaviatūros įvedamas tekstas yra nukreipiamas į failą **tekstas3**. Įvedamas tekstas rašomas eilutėmis (**alfa beta**).

Teksto įvedimas baigiamas naujoje eilutėje, kartu paspaudus klavišus **Ctrl ir D**. Šių klavišų kombinacija LINUX sistemoje reiškia srauto pabaigą.

4.3.1.6. Failo peržiūros komandos more/less

Komandos skirtos ilgiems failams peržiūrėti. Komanda

```
more tekstas3
```

išves failo **tekstas3** turinį puslapiais. Apačioje matyti užrašas:

```
--More-- (nn%)
```

Vietoj **nn** yra skaičiai, kurie parodo, kiek dokumento procentų jau yra išvesta ekrane. Norint pamatyti kitą puslapį, reikia spausti *space* klavišą. Norint pamatyti vieną tolesnę eilutę, galima spausti *Enter* klavišą. Klavišu *q* galima nutraukti failo peržiūrą, klavišas *b* persuka dokumentą per vieną ekrano ilgį atgal.

Komanda **less** turi tam tikrų savybių, kuriomis ji pranašesnė už komandą **more**, nes **less** komanda gali būti pritaikoma vartotojo reikmėms. Kai kurios **less** komandos yra panašios į redaktoriaus **vi** komandas. Failo peržiūra nustatoma komanda

```
less failas
```

Ji parodys pirmą failo puslapį, kairiajame apatiniame kampe parodys failo vardą. Tolesniems veiksams su failu atlikti galima naudoti **less** komandas. Veiksmai su komanda **less** baigiami paspaudus „q“ klavišą. Jei dar neturite failų, tai galite panaudoti šias komandas sisteminių failų peržiūrai:

```
more /etc/passwd
```

```
more /proc/version
```

4.3.1.7. Komanda tail/head

tail komanda skirta paskutinėms failo eilutėms peržiūrėti. Esant nutylėjimui, išvedamos paskutinės 10 failo eilučių. Taigi, norėdami pamatyti failo paskutines eilutes, galime naudoti komandą

```
tail /etc/passwd
```

Komanda **tail** gali būti naudojama ir su išvedimo pradžios apibrėžimu, pavyzdžiui:

```
tail +20 failas
```

```
tail -20 failas
```

Pirmuoju atveju išvedimas pradedamas nuo 20 failo eilutės, antruoju atveju - nuo 20 eilutės nuo failo galo.

Komanda **head** yra priešinga komandai **tail**. **head** naudojama pirmoms failo eilutėms išvesti. Esant nutylėjimui, išvedamos pirmos 10 failo eilučių. Taigi galima nurodyti išvedamų nuo failo pradžios eilučių skaičių, pavyzdžiui,

```
head -20 /etc/passwd
```

Šiuo atveju bus išvestos pirmos 20 eilučių.

4.3.1.8. Failų kopijavimo komanda cp

Norint panaudoti šią komandą, galimi du atvejai:

- galima kopijuoti failą iš bet kurio kito vartotojo katalogo į savo namų katalogą.

Tokios komandos sintaksė yra tokia:

```
cp 'kelias iki failo/failo vardas' 'kelias iki išsaugojimo vietos/failo vardas'
```

Pavyzdžiui, vartotojas, kurio prisijungimo vardas yra **user2**, nori kopijuoti failą **pirmas.c**, esantį vartotojo **user1** kataloge **lab1**, į savo namų katalogą tuo pačiu vardu (tarkim; kad, vartotojo **user1** namų katalogas yra **/home/user1**, vartotojo **user2** namų katalogas yra **/home/user2**):

Galima naudoti vieną iš tokių šios komandos variantų (3, 4, 5 variantai galimi ne visuose apvalkaluose):

```
cp /home/user1/lab1/pirmas.c    /home/user2/pirmas.c
cp /home/user1/lab1/pirmas.c    /home/user2/
cp /home/user1/lab1/pirmas.c    ~/pirmas.c
cp /home/user1/lab1/pirmas.c    ~/
cp ~/user1/lab1/pirmas.c        /home/user2/
```

Galima kopijuoti failą einamajame kataloge, išsaugant jį kitu vardu:

```
cp pirmas.c    antras.c
```

Bus gauta failo **pirmas.c** kopija vardu **antras.c**.

4.3.1.9. Failo perkėlimo komanda mv

Komanda **mkdir** yra skirta naujam katalogui sukurti. Ja naudojantis, galima sukurti naują pakatalogį einamajame kataloge. Pavyzdžiui:

```
mkdir first
```

Bus sukurtas pakatalogis **first** einamajame kataloge. Norint tuo įsitikinti, galima pasinaudoti komanda **ls**.

Pakalbėkime apie pačią komandą **mv**.

Ji gali būti reikalinga failo vardui pakeisti. Pavyzdžiui:

```
mv pirmas antras
```


Šiuo atveju vietoje einamajame kataloge buvusio failo **pirmas** atsiras failas **antras**. Iš tikrųjų failų sistemoje failas nebus kilojamas, bus pakeistas tik jo vardas.

Pervardijimą galima atlikti su užklausa:

```
mv -i pirmas antras  
remove pirmas?n
```

Prieš perkeltiant, sistema užklauso, ar iš tikrųjų norite perkelti failą. Galima paspausti klavišus **y** (taip) arba **n** (ne). Taigi **n** įvedimas reiškia, kad failo perkėlimas yra nutraukiamas.

Keliant failą į kitą katalogą, reikia nurodyti, į kurį katalogą failas bus keliamas. Tarkime, failą **pirmas** reikia kelti į einamojo katalogo pakatalogį **first** tuo pačiu vardu:

```
mv pirmas first/pirmas
```

arba:

```
mv pirmas first/
```

4.3.1.10. Failo ištrynimo komanda **rm**

Komanda **rm** skirta failui ištrinti. Failas dingsta ir jokios galimybės jo atkurti nėra. Norėdami apsisaugoti, bandykite naudoti trynimo komandą su užklausa **rm -i**.

Pavyzdžiui:

```
rm -i antras  
rm -i first/pirmas  
rm -i first/*.c
```

Antroji komanda trins failą **pirmas**, esantį pakatalogyje **first**. Trečioji komanda iš pakatalogio **first** trins visus failus, turinčius bet kokį vardą ir priesagą **c**.

Komanda **rm *** ištrins failus, turinčius bet kokius vardus.

4.3.1.11. Vartotojų teisės ir jų keitimas **chmod** komanda

Failams-duomenims ir programoms saugoti nuo sugadinimo, neautorizuoto, neleistino naudojimo Linux naudoja prieinamumą (leidimų) mechanizmą (*permissions*). Kiekvienas Linux failas (Linux beveik viskas yra failai) yra susietas su tam tikru savininku bei grupe. Vartotojas, sukūręs failą, tampa jo savininku, o failo priklausomybę grupei nusako šio vartotojo grupė. Po to failo savininkas gali būti pakeičiamas, naudojant komandą **chown**, o priskyrimas grupei pakeičiamas naudojant komandą **chgrp**.

Baziniai prieinamumai, susieti su failu, yra tokie:

read (leidimas skaityti duomenis iš failo);

write (leidimas keisti failą);

execute (leidimas vykdyti jį kaip programą).

Kiekvienas failas turi po tris šių prieinamumų grupes: pirmą, nusakančią leidimus failo savininkui, antrą, nusakančią prieinamumus tos pačios grupės nariams, kuriai ir priklauso failas, ir trečią, nusakančią prieinamumus visiems kitiems sistemos vartotojams. Šias privilegijas vartotojas įgyja prisijungęs prie sistemos. Jis gali skaityti, rašyti ir vykdyti jam priklausančius failus, turinčius atitinkamus leidimo bitus. Jis gali atlikti šiuos veiksmus su failais, turinčiais nustatytus leidimus šio vartotojo grupei. Taip pat jis gali atlikti veiksmus su failais, kurie pagal nustatytus leidimus yra pasiekiami visiems sistemos vartotojams.

Leidimų nustatymus galima pamatyti, išvedus katalogo turinį, kai naudojama ilgojo išvedimo komanda

ls -l

Tarkime, kad ši komanda, pritaikyta tik vienam failui **notes**, išvedė:

```
ls -l notes
-rw-r--r--  1 marsav      users          2993 Jun 17 11:00 notes
```

notes - tai yra paprastas duomenų failas. Iš išvestos informacijos matyti, kad šio failo savininkas yra vartotojas **marsav** ir šis failas yra sukurtas, priklausant grupei **users**. Kitos populiarios vartotojų grupės yra **staff**, **admin**, **guest**, **student** ir kt. Grupės sąvoka nėra labai svarbi, jei mašina naudojasi vienas vartotojas, ji tampa svarbi tada, kai mašina naudojasi daug skirtingas privilegijas turinčių vartotojų. Simboliai **-rw-r--** atspindi failo **notes** prieinamumus.

Kairysis brūkšnelis rodo, kad tai yra paprastas failas, jei tai būtų katalogas, čia matytume raidę **d**. Raidė **l** žymi simbolinį ryšį, **c** arba **b** – specialųjį failą, kuris aprašo įrenginį. Raidė **c** žymi tai, kad duomenų mainai galimi simboliais, o **b** žymi tai, kad šie mainai galimi blokais.

Kiti trys simboliai **rw-** žymi prieinamumus šio failo savininkui-vartotojui, kurio prisijungimo prie sistemos vardas yra **marsav**. Failo savininkas gali failą skaityti arba rašyti (keisti). Failo savininkas negali šio failo vykdyti, nes trečias simbolis šioje sekoje yra **-**.

Kitų trijų simbolių grupė **r--** žymi leidimus **users** grupės nariams. Jie gali šį failą tik skaityti.

Paskutinė trijų simbolių grupė rodo leidimus visiems kitiems sistemos vartotojams. Jie taip pat šį failą gali tik skaityti. Tai gana dažnai pasitaikanti leidimų (prieinamumų) kombinacija.

Panagrinėkime kitą pavyzdį, kur **gcc** yra GNU C kalbos kompiliatorius.

```
ls -l /usr/bin/gcc
-rwxr-xr-x  3 root      bin          64796 Mar 21 16:41 /usr/bin/gcc
```

Failas priklauso vartotojui **root**, grupei **bin**. Į jį rašyti, tai yra jį modifikuoti, gali tik failo savininkas **root**, tačiau šis failas gali būti nuskaitomas ir vykdomas visų. Šitokie leidimai yra nustatomi Linux komandoms.

Root vartotojas - tai vartotojas, turintis sistemoje **0** identifikatorių (ID). Tai specialus, administratoriaus prisijungimas, kuriam sistemoje leidžiami visi veiksmai ir kuris turi visas įmanomas privilegijas. Šis prisijungimas yra svarbus sistemos administravimui. Tačiau tai ir labai pavojingas prisijungimas. Administratoriaus klaidos (arba šio prisijungimo slaptažodį sužinojusio įsilaužėlio veiksmai) gali padaryti daug žalos sistemai.

Kadangi **root** prisijungimas yra pavojingas sistemos saugos požiūriu, neteisėti priėjimai prie **root** prisijungimo įvaldymo sistemoje turi būti neleistini. Tai svarbu saugant **root** slaptažodį, kuris yra tiesioginis įsilaužėlių atakų taikiny.

Trumpai pakalbėsime apie slaptažodžius, jų saugą. Paprastas slaptažodis palengvina įsilaužėlių veiksmus: atspėję jūsų slaptažodį, jie gali pasinaudoti jūsų prisijungimu, jums suteiktomis sistemoje privilegijomis, visi įsilaužėlių veiksmai bus atliekami jūsų vardu. Taigi, nereikia padėti įsilaužėjams, parenkant lengvai atspėjamus slaptažodžius. Įsilaužėjai atspėdami („nulaūždami“) slaptažodžius remiasi ir žodynais.

Todėl į slaptažodį reikėtų įtraukti skaičių, specialių ženklų. Tuomet slaptažodis taptų sunkiau atspėjamas.

Panagrinėkime trečią pavyzdį:

```
ls -ld ~
drwxr-xr-x  89 esr      users          9216 Jun 27 11:29 /home2/esr
```

Šiuo atveju failas yra katalogas. Tai rodo raidė **d**, pateikta pirmu simboliu. Matyti, kad į katalogą gali rašyti tik vartotojas **esr**, bet jį skaityti ir vykdyti gali visi vartotojai.

Leidimas skaityti katalogą reiškia, kad galima peržiūrėti katalogo turinį, t.y. matyti failų bei pakatalogių, esančių tame kataloge, vardus.

Leidimas rašyti suteikia teisę šiame kataloge kurti naujus failus arba juos išmesti.

Leidimas vykdyti katalogo atžvilgiu reiškia tai, kad galima pasiekti žemiau šio katalogo esančius pakatalogius bei failus.

Jei katalogas turės vykdymo leidimą visiems vartotojams, bet neturės skaitymo leidimo visiems vartotojams, tai bet koks vartotojas galės pasiekti žemiau šio katalogo esančius failus, jei žinos šių failų vardus (nes peržiūrėti katalogo turinio nebus leista).

Rašymo į katalogą leidimas reiškia tik tai, kad vartotojas gali kurti failą arba išmesti jį iš katalogo. Tačiau tai nereiškia, kad jis gali rašyti į kurį nors katalogo failą.

Dar vienas pavyzdys su programos **login** leidimais:

```
ls -l /bin/login
-rwsr-xr-x  1 root      bin           20164 Apr 17 12:57 /bin/login
```

Leidimų eilutė yra tokia pati kaip ir leidimų kombinacija, skirta vykdomoms komandoms. Skirtumas tas, kad savininko leidimuose vietoje raidės **x** yra raidė **s**. Tokia leidimo reikšmė reiškia, kad, vykdant šią programą, bus keičiamas vartotojo efektyvusis ID. Šis leidimo bitas yra vadinama **setuid** bitu.

Šitoks **setuid** bitas paprastai taikomas toms sisteminiams komandoms, kurios „nurodo“ suteikti paprastiems vartotojams **root** privilegijų, kad jie galėtų atlikti veiksmus su tam tikrais sistemos duomenimis. Tačiau tai turi būti daroma, kontroliuojant šiuos veiksmus. Jei programai šis bitas yra nustatytas, tai jos vykdymo metu vartotojas įgyja programos savininko privilegijas.

Tiek **root** prisijungimas, tiek **setuid** programos yra naudingos, bet kartu ir pavojingos. Kiekvienas, kuris gali modifikuoti **root** priklausančią programą su nustatyta **setuid** reikšme, gali sugebėti paleisti apvalkalo procesą su **root** privilegijomis. Dėl šios priežasties daugumoje Linux sistemų, kai tik šis failas atidaromas rašyti, jam automatiškai išjungiamas **setuid** bitas. Daugelis Linux atakuotojų bando pasinaudoti programų su nustatytais **setuid** bitais spragomis, kad galėtų įgyti **root** teises. Dėl to sistemos administratorius turi atidžiai stebėti šias programas - ar neatsirado naujų programų su nustatytais **setuid** bitais.

Kalbėdami apie failų, katalogų prieinamumus (leidimus) nepaminėjome, kaip jie yra nustatomi naujai sukuriams failams. Linux sistemos vartotojas gali būti kelių grupių narys. Tai leidžia jam suteikti skirtingus leidimus veiksmams su skirtingais failais. Tačiau vartotojas prisijungęs prie sistemos, yra tam tikros grupės narys ir vartotojo naujai sukurtiems failams priskiriama ta grupė.

Leidimų nustatymas naujai sukurtiems failams ar katalogams yra susijęs su vartotojo aplinkos kintamuoju **umask**. Šis kintamasis nulemia, kurie leidimo bitai turi būti išjungiami, sukuriant naują failą.

Dažnai naudojama tokia kintamojo **umask** reikšmė:
022; arba raidėmis tai atitiktų seką **- - - -w- -w- .**
Taigi, naujai sukurtas failas turėtų tokius leidimus:
rwxr-xr-x.

chmod komanda keičia failo naudojimo teises (mode). Šiuos požymius gali keisti tik savininkas savo asmeniniame (home) kataloge. Failo naudojimo teises galima sužinoti, įvedus komandą **ls -l**. Komandos formatas:

chmod [kas] operatorius [teisė(s)] failo vardas

kas – tai 0 ar daugiau simbolių, rodančių, kuriems vartotojams yra keičiamos teisės į failą.

Simboliai gali būti	Operatorius gali būti	Teisės gali būti:
u - savininko teisės, g - savininko grupės teisės, o - kitų vartotojų teisės, a - visų vartotojų teisės;	+ - suteikti teisę, - - panaikinti teisę, = - priskirti teisę	r - skaityti failą, w - rašyti į failą, x - vykdyti programą.

Failo vardas - tai failas (ar katalogas), kurio naudojimo teisės yra keičiamos.

Pavyzdžiai:

chmod u+x programa

Vartotojui - failo **programa** savininkui(u) suteikiama (+) teisė vykdyti (x) failą **programa**.

chmod u+rw, g-x programa

Savininkui (u) suteikiamos (+) visos teisės į failą **programa** (teisė skaityti šį failą (r), teisė rašyti į šį failą (w), teisė vykdyti (x)), o vartotojo grupė (g) netenka teisės (-) vykdyti (x) failą **programa**.

chmod a+r, go-wx mano

visiems vartotojams (a) suteikiama teisė (+) skaityti r failą, o grupė (g) ir kiti vartotojai (o) netenka (-) teisės rašyti (w) ir vykdyti (x) failą **mano**.

4.3.2. Informacinės komandos

date data, laikas;
uname informacija apie sistemą;
uptime informacija apie sistemos perkrovimą.

Pavyzdžiui:

date
Fri Jan 26 11:41:40 EET 2007
uptime
11:41am up 2:20, 2 users, load average: 0.00, 0.00, 0.00

4.3.3. Informacija apie vartotoją

whoami išveda vartotojo prisijungimo vardą;

users	išveda su vietine mašina dirbančių vartotojų sąrašą;
rusers	išveda vietiniame tinkle dirbančių vartotojų sąrašą;
logname	išveda vartotojo prisijungimo vardą;
id	vartotojo ID bei vartotojo grupės ID;
groups	išveda, kurioms grupėms priklauso vartotojas;
who	išveda informaciją apie su vietine mašina dirbančius vartotojus;
finger	registracinis įrašas apie vartotoją;
last	informacija apie vartotojo darbą sistemoje.

Pavyzdžiui:

```

who
saule      pts/2      Jan 26 11:32      (c1-02.pit.ktu.lt)
whoami
saule
id
uid=104(saule) gid=1000(stud)

```

Čia komanda **who** rodo, kad šiuo metu su sistema dirba vartotojas **saule**, rodo jo prisijungimo terminalą, prisijungimo laiką bei nutolusio prisijungimo kompiuterio adresą. Komanda **whoami** rodo vartotojo prisijungimo vardą, o komanda **id** vartotojo identifikacinį numerį bei jo grupės identifikacinį numerį sistemoje.

4.3.4. Komanda **ps**

Ši komanda parodo informaciją apie šiuo metu sistemoje vykstančius procesus bei vykdomų procesų būklę. Jei komandoje nenurodomos jokios parinktys, išvedami jūsų paleisti procesai. Tai yra išvedama informacija apie einamojo apvalkalo paleistus procesus, spausdinama informacija apie tuos procesus, kurių vartotojo efektyvus ID ir kontroliuojamas terminalas yra toks pat kaip ir iškvietusio **ps** komandą. Pavyzdžiui:

```

ps
PID TTY      TIME CMD
1830 pts/1    0:00 tcsh
1859 pts/1    0:00 ps

```

Šiame išvedime pateikta informacija apie du vartotojo procesus: vartotoją aptarnaujančią apvalkalo procesą **tcsh**, bei procesą **ps**. Stulpelyje **PID** yra nurodomas proceso numeris, **TTY**-terminalas, **TIME** – kiek laiko buvo vykdytas procesas, **CMD** – proceso vardas.

Jei **ps** komandoje nurodytos parinktys, tai išvedimas nusakomas panaudotomis parinktimis. Komandoje **ps** galima panaudoti kelias parinktis ir jas sujungti.

Kai kurių **ps** komandos parinkčių prasmė:

Parinktis Prasmė

- A** Išveda visus procesus (įskaitant ir kitų vartotojų);
- a** Išveda visus procesus, susijusius su naudojamu **tty**, išskyrus sesijos lyderius;
- d** Išveda visus procesus, išskyrus sesijos lyderius;
- e** Išveda visus šiuo metu vykdomus procesus;
- l** Generuoja ilgą išvedimą;
- o** Teikia galimybę tvarkyti išvedimą.

Pavyzdžiui:

```
ps
  PID TTY          TIME CMD
 1482 pts/1        0:00 tcsh
 1546 pts/1        0:00 ps
 1411 pts/1        0:00 tcsh
```

```
ps -a
  PID TTY          TIME CMD
 1482 pts/1        0:00 tcsh
 1547 pts/1        0:00 ps
```

```
ps -l
 F S  UID        PID  PPID  C PRI  NI           ADDR           SZ        WCHAN  TTY          TIME CMD
 8 S  2011        1482  1411   0  50  20  70125bc8       297  70c59ec6 pts/1        0:00 tcsh
 8 0  2011        1549  1482   0  50  20  70caf090       123                pts/1        0:00 ps
 8 S  2011        1411  1409   0  40  20  70cae9a8       450  70baeb26 pts/1        0:00 tcsh
```

Komanda **ps** rezultatus išveda stulpeliais, kurių prasmė tokia:

F

Šiame stulpelyje spausdinama su procesu susijusios vėliavėlės reikšmė (istoriškai naudojama, tačiau specialios prasmės neteikia).

S

Proceso būvis, kuris nusakomas raidėmis:

- D** Procesas, kuris „miega“ ir laukia I/O;
- S** „Miegantis“ procesas – jis laukia kokio nors įvykio pasirodymo;
- R** Vykdomas procesas (arba esantis pasirengusių procesų eilėje);
- Z** Procesas-„zombis“. Procesas pasibaigęs, o tėvas jo nelaukia;
- T** Procesas yra sustabdytas arba trasuojamas.

UID

Proceso vartotojo efektyvus ID numeris.

PID

Proceso ID numeris (jis reikalingas, norint nutraukti procesą).

PPID

Šio proceso tėvo proceso ID numeris.

C

Procesoriaus panaudojimas dispečeriavimui (tai pasenusi parinktis). Nėra išvedama, naudojant **-c** parinktį **ps** komandoje.

CLS

Dispečeriavimo klasė. Išvedama **ps** komandoje, naudojant **-c** parinktį.

PRI

Proceso prioritetas. Nenaudojant **-c** parinkties, didesnės reikšmės rodo mažesnę prioritetą. Naudojant **-c** parinktį, didesni skaičiai reiškia didesnę prioritetą.

NI

Tai kintamojo **Nice** reikšmė, kuri naudojama, skaičiuojant proceso prioritetą.

ADDR

Proceso adresas atmintyje.

SZ

Bendras proceso virtualios atminties dydis puslapiais, įskaitant ir sritį, susijusią su proceso failais ir įrenginiais.

WCHAN

Adresas įvykio, kurio procesas laukia, būdamas „miego“ (*sleeping*) būsenoje, (tuščia, jei procesas yra vykdomas).

STIME

Proceso pradžios laikas pateikiamas valandomis, minutėmis ir sekundėmis (jei procesas yra prasidėjęs daugiau nei prieš 24 valandas, pateikiamas mėnuo ir diena).

TTY

Proceso kontroliuojamas terminalas (? Spausdinama, jei procesas nesusietas su terminalu).

TIME

Kumuliacinis (visas gautas iki to momento) proceso vykdymo laikas.

CMD

Komandos vardas.

Du papildomi stulpeliai spausdinami naudojant **-j** parinktį:

PGID

Procesų grupės lyderio (vadovaujančiojo proceso) ID.

SID

Sesijos lyderio proceso ID

Naudojant **-o** parinktį **ps** komandoje leidžiama vartotojui pačiam nusakyti išvedamos informacijos formatus. Bendra komandos forma yra

ps -o format

Formatas apibrėžiamas standartiniais vardais, nusakančiais, kokius stulpelius vartotojas nori išvesti. Standartinius vardus atitinkanti informacija bus išvesta tokia tvarka, kokia ji nurodyta **ps** komandoje. Pavyzdžiui:

ps -o pid,ppid

PID PPID

1810 1808

1817 1810

4.4. Linux apvalkalas ir programų vykdymas apvalkalo aplinkoje

Apvalkalas (*shell*) – tai Linux komandų interpretatorius. Jis interpretuoja įvedamas komandas. Vadinamas apvalkalu, nes apgaubia ir paslepia operacinės sistemos branduolį. Linux apvalkalas ir branduolys yra skirtingos, atskiros programos, kurios komunikuoja viena su kita, naudodamos nedidelį sisteminių kreipinių rinkinį. Vartotojui pasirinkti yra keletas apvalkalų: **sh**, **cs****h**, **ksh**, **tcsh**, **bash**, ... Jie skiriasi tarpusavyje savo naudojama sintakse bei galimybėmis.

Vartotojo klaviatūra įvedamas tekstas kartu yra išvedamas ekrane, o paspaudus **Enter**, įvesta eilutė perduodama apvalkalo procesui. Apvalkalo procesas bando interpretuoti įvestą eilutę kaip komandą. Tarkime, norint pamatyti einamojo katalogo turinį, įvedama **ls** komanda. Apvalkalo procesas pritaiko atitinkamas taisykles, interpretuodamas įvestą tekstą, bei išsiaiškina, kad reikia įvykdyti vykdomąjį modulį **/bin/ls**. Apvalkalas kreipiasi į operacinės sistemos branduolį, prašydamas paleisti **/bin/ls** kaip apvalkalo proceso naują vaiko procesą. Šio naujo proceso išvedimas yra susiejamas su ekranu, o įvedimas - su klaviatūra. Kol yra vykdomas šis vaiko procesas (**ls**), apvalkalo procesas „miega“. Kai **/bin/ls** proceso veiksmas baigti, yra vykdomas sisteminis kreipinys **exit**, kuriuo branduoliui yra pranešama apie proceso pabaigą. Branduolys apie tai informuoja apvalkalo procesą, kuris gali tęsti savo veiksmus toliau. Apvalkalo procesas į

ekraną vėl išveda kreipinį (pavyzdžiui, \$ simbolį **sh** apvalkalo atveju). Laukiama naujos vartotojo komandos įvedimo.

Apvalkalas yra komandų interpretatorius, kuris priima kiekvieną vartotojo įvedamą komandą ir perduoda ją įvykdyti operacinės sistemos branduoliui. Įvykdžius komandą, apvalkalas išveda komandos rezultatus arba pranešimus apie klaidas į ekraną ir laukia kitos komandos.

Sistemos administratorius, registruodamas naują vartotoją, jam aptarnauti skiria kurį nors iš sistemoje naudojamų apvalkalų. Kiekvienas apvalkalas turi ir savas komandas, kurios skiriasi savo rašymo taisyklėmis.

Apvalkalo interpretuojamose komandose leidžiama perskirstyti įvedimo, išvedimo arba klaidų srautą. Standartiniu įvedimo failu yra laikomas failas **stdin**, kuris yra susiejamas su įvedimu klaviatūra. Standartiniu išvedimo failu laikomas failas **stdout**, susiejamas su išvedimu į ekraną. Standartiniu klaidų failu laikomas failas **stderr**, kuris, esant nutylėjimui, taip pat yra susiejamas su išvedimu į ekraną. Srautus galima nukreipti kitur, pavyzdžiui, standartinį išvedimą galima nukreipti į failą, įvedamus duomenis „imti“ ne iš klaviatūros, o iš failo. Nukreipiama naudojant šiuos simbolius:

- > stdout perskirstymas (užrašymas ant viršaus į nurodomą failą);
- >> stdout perskirstymas (pirašymas failo gale – *append* operacija);
- < stdin perskirstymas;
- 2> stderr perskirstymas (sh, ksh, bash apvalkalų atveju);
- >& stdout ir stderr perskirstymas (csh, tcsh apvalkalų atveju).

Pavyzdžiui:

who > ss

Šiuo atveju komandos **who** išvedimas yra rašomas į failą **ss**. Ekrane jo nematyti.

„Vamzdžio“ (*pipe*) mechanizmas yra naudojamas apvalkale vienos komandos **stdout** failui perduoti kaip **stdin** failą kitai komandai. Komandos sintaksė yra tokia:

command1 [arguments] | command2 [arguments]

Pavyzdžiui:

ps -aef | grep sshd | grep -v root

Šiuo atveju komandos **ps -aef** išvedimas bus nukreiptas kaip duomenų failas komandai **grep sshd**, kuri filtruos eilutes, palikdama tik eilutes apie **sshd** procesus. Filtravimo rezultatus gaus kita komanda **grep -v root**, kuri filtruodama (komanda **grep**) atmes eilutes (parinktis **v**) su **root** valdomais procesais.

Vartotojui įvedus komandą apvalkalas tikrina ją ir iš pradžių apdoroja rastus metasimbolius, o tada vykdo suformuotą komandą.

Metasimboliai ir jų reikšmės pateikti lentelėje.

Simbolis	Reikšmė
*	Failo vardų metasimbolis; atitinka nulinį arba didesnį bet kokių simbolių skaičių
?	Failo vardų metasimbolis; atitinka vieną bet kokią simbolį
[...]	Failo vardų metasimbolis; atitinka vieną bet kokią simbolį iš skliaustuose nurodytų simbolių
`komanda`	Įkelia nurodytos komandos įvykdymo rezultatus
;	Komandų skyriklis
	Sąlyginis komandų vykdymas; komanda vykdoma, jei prieš ją esanti komanda nebuvo įvykdyta sėkmingai
&&	Sąlyginis komandų vykdymas; komanda vykdoma, jei

	prieš ją esanti komanda buvo įvykdyta sėkmingai
(...)	Komandų grupavimas
&	Komanda vykdoma foniniu režimu
#	Visi simboliai iki eilutės galo interpretuojami kaip komentarai
\$	Kintamojo reikšmė

Pavyzdžiui:

```
more *.c
```

Ši komanda skaitys visus einamojo katalogo failus, kurie turi priesagą **.c** (C kalbos programas). Jei tokių failų komanda neras, bus išspausdintas pranešimas apie klaidą:

```
*.c: No such file or directory
```

```
more pav[a-d].c
```

Šiuo atveju bus parodomas failų, kurių vardai yra **pava.c**, **pavb.c**, **pavc.c** arba **pavd.c** turinys, jei jie yra einamajame kataloge.

```
echo `ls /var`
```

```
adm apache apache2 appserver audit ...
```

Šiame pavyzdyje panaudota komanda **echo**, kuriai kaip argumentas yra nurodytas komandos **ls** išvedimas. Apvalkalas, prieš vykdydamas komandą **echo**, įvykdo komandą **ls**, nurodytą tarp kairinių kirčių, ir komandos **ls** rezultatus pateikia komandai **echo** kaip argumentus.

Paprastai komandos rašomos po vieną. Norint užrašyti kelias komandas vienoje eilutėje, jos yra atskiriamos simboliu „;“. Pavyzdžiui:

```
cd ../; pwd
```

Pagal šias komandas bus pereita į vienu lygiu aukštesnį katalogą ir išspausdintas visas kelio vardas iki to katalogo.

```
cd rr || cd lab2
```

```
cd: rr: No such file or directory
```

```
pwd
```

```
/home/os/lab2
```

Šiame pavyzdyje prašoma pereiti į katalogą **lab2**, jei nepavyksta pereiti į katalogą **rr**.

```
cd lab2 && ls
```

```
awk      failai      grepai      rusiuot      sablonai
```

Šioje konstrukcijoje komanda **ls** bus vykdoma tuo atveju, jei pavyks įvykdyti komandą **cd**.

Veiksmams, kurie užrašyti skliaustuose, atlikti paprastai yra iškviečiamas kitas apvalkalo procesas, pavyzdžiui:

```
pwd ;(cd lab2; pwd); pwd
```

```
/home/os
```

```
/home/os/lab2
```

```
/home/os
```

Čia skliaustuose yra nurodoma pereiti į katalogą **lab2** ir išspausdinti visą kelią iki šio katalogo. Kadangi veiksmus skliaustuose vykdo kitas apvalkalo procesas, tai, skliaustams užsidarius, esame tame pačiame kataloge, kuriame buvome prieš skliaustus.

Jei simbolis **&** yra užrašytas komandos gale, tai jis liepia komandą vykdyti foniniu režimu. Pavyzdžiui:

```
sleep 10&
```

[1] 24917

jobs

[1]+ Running

sleep 10 &

Komanda **sleep** šiuo atveju bus paleista foniniu režimu.

Metasimboliai #, \$ paprastai naudojami rašant apvalkalo procedūras (*scripts*).

Kabutės naudojamos apvalkalui apibrėžti, koks parengiamasis darbas turi būti atliktas, prieš pradedant vykdyti pačią komandą:

- viengubos kabutės draudžia keisti eilutėje esančius metasimbolius atitinkamais vardais, pateikti kintamųjų reikšmes, komandų vykdymo rezultatus; t. y. panaikina visų apskliaustų ženklų specialią prasmę;
- dvigubos kabutės panaikina metasimbolių specialią prasmę, išskyrus \$ ` ir \

Kiekvienas apvalkalas turi rinkinį rezervuotų aplinkos kintamųjų, kurių reikšmės inicijuojamos vykdant prisijungimo failus (pvz., **.cshrc**). Be to, kiekvienas apvalkalas turi rezervuotą specialios paskirties vietinių kintamųjų rinkinį. Kiti aplinkos bei vietiniai kintamieji gali būti vartotojo sukuriama ir naudojami.

Aplinkos kintamųjų nustatytas reikšmės galima sužinoti komanda **env** (**sh** apvalkalo atveju). Pavyzdžiui, **sh** apvalkalo atveju yra:

env

HOME=/home/saule

LOGNAME=saule

MAIL=/var/mail/saule

PATH=/bin

SHELL=/bin/sh

Norint sužinoti atskirai paimto vieno kintamojo reikšmę, naudojama komanda **echo**, pavyzdžiui:

echo \$PATH

/usr/bin

Apvalkalo komandos gali būti rašomos į failus, o po to šie failai vykdomi. Tokie failai kartais yra vadinami apvalkalo procedūromis (*shell scripts*). Koks apvalkalas bus kviečiamas šio failo vykdymui gali būti nurodoma arba faile, arba nurodant apvalkalą komandų eilutėje, prieš failo vardą.

Pavyzdžiui, turime procedūrą **procb**, kurioje įrašytos šios komandos.

```
#!/bin/sh
num=0
for param in * #su kiekvienu katalogo failu
do
    num=`expr $num + 1`
    echo failas $num yra tokio dydžio
    wc -c $param
done
```

Jos paleidimo komanda ir rezultatai:

chmod u+x procb

./procb

failas 1 yra tokio dydžio

1877 4659.tgz

failas 2 yra tokio dydžio

2 aa

failas 3 yra tokio dydžio

2694 aaa
failas 4 yra tokio dydžio
11 ab

Apvalkalo procedūroms rašyti reikia žinoti atitinkamo apvalkalo komandas, tačiau parašytos procedūros tampa tarsi naujomis komandomis, kuriomis sistemos vartotojas gali papildyti standartinių komandų rinkinį.

Literatūra:

N. Sarafinienė, K. Pocius. UNIX: Mokomoji knyga.-Kaunas: Technologija, 2006, ISBN 9955-09-991-7. 80 p.

Linux sistemos administravimas: [antrojo leidimo vertimas]/Vicki Stanfield, Roderick W. Smith.- Kaunas : Smaltija,2003.