

## OS KONSPEKTAS K1

### Turinys

L1/1 .....	3
Parinktys .....	3
Operandai .....	3
Vykdytų komandų panaudojimas .....	4
man puslapiai .....	4
man -k ir apropos .....	5
info puslapiai .....	5
Kelio vardas ( <i>path</i> ) .....	6
Darbinis katalogas .....	7
Tipiniai katalogai ir jų paskirtis .....	7
pwd - darbinio katalogo sužinojimas .....	7
cd - darbinio katalogo keitimas .....	7
ls - katalogo turinio peržiūra .....	7
find - paieška .....	7
stat - detali informacija apie failą .....	8
touch - naujo tuščio failo kūrimas arba datų pakeitimas .....	8
mkdir - naujo katalogo sukūrimas .....	8
rmdir - tuščio katalogo naikinimas .....	8
cp - failo kopijavimas .....	8
mv - failo ar kataogo pervadinimas ir/arba perkėlimas į kitą vietą .....	8
rm - failo ar katalogo trynimas .....	9
ln - nuorodų kūrimas .....	9
L1/2 .....	11
more .....	11
head .....	12
tail .....	12
cat .....	12
Nano redaktorius .....	12
grep .....	14
file .....	14
strings .....	14

diff .....	15
uniq .....	15
od .....	16
wc .....	16
iconv .....	17
sort .....	17
split .....	18
cmp .....	18
cksum .....	18
L1/3 .....	19
Paslėpti failai .....	21
id – naudotojo identifikatorių nustatymas .....	21
groups – grupių, kurioms priklauso naudotojas sąrašas .....	22
getent – OS duomenų bazių turinio peržiūra .....	22
ls – katalogo turinio peržiūra (detaliau) .....	22
chmod – leidimų keitimas .....	23
umask – naujai kuriamų failų leidimai .....	24
du – užimama vieta .....	24
df – laisva vieta .....	24
L1/4 .....	25
Procesų medis .....	25
Procesų grupė .....	25
Sesija .....	26
Saugumo sistema .....	26
Proceso atributai .....	26
Laiko apskaita .....	27
Prioritetas .....	27
ps - procesų sąrašas .....	27
nice, renice - proceso prioriteto keitimas .....	29
top - interaktyvus procesų sąrašas .....	30
strace - proceso iškviečiami <i>syscall'ai</i> .....	31
kill - nutraukti procesą .....	31
^Z, fg, bg, jobs, kill – komandos užmigdytas ir žadinimas .....	32
who – prie sistemos prisijungusių naudotojų sąrašas .....	33

iostat – I/O statistika.....	34
vmstat - virtualios atminties statistika.....	34
mpstat - kiekvieno CPU statistika .....	35
sar - sistemos apkrovimo statistika .....	35

## L1/1

<i>Enter</i>	komandos įvedimo pabaiga (po Enter paspaudimo įvesta komandos eilutė vykdoma).
<i>^C (Ctrl-C)</i>	komandos nutraukimas
<i>^D</i>	"failo pabaiga", t.y. jei komanda laukia duomenų iš terminalo įvedimo, naujoje eilutėje paspaudus <i>^D</i> galima nurodyti, kad daugiau duomenų nebus
<i>^L</i>	ekrano valymas
<i>^Z</i>	komandos "užmigdymas" ( <b>komanda nenutraukiama</b> tik sustabdoma)
<i>^S</i>	terminalo įveimo/išvedimo užmigdymas (terminalo langas nebereaguoja į klavišų paspaudimus ir neberodo kas išvedama, bet šia informaciją atsimena)
<i>^Q</i>	terminalo "pažadėjimas" (jei jis buvo sustabdytas su <i>^S</i> – atvaizduojama „užlaikyta“ informacija)

komanda [-parinktys(-ys)] [operandas (-ai)]

**Parinktys** (*angl. options, flags*) - **komandos argumentai**, leidžiantys keisti (išplėsti/sumažinti) komandos funkcionalumą. Paprastai prieš parinktį yra dedamas „-“ ženklas. Vedant komandą galima nurodyti vieną, dvi ir daugiau parinkčių, jas apjungti ir pan.

**Operandai** – **komandos argumentai**, leidžiantys nurodyti objektus, kuriais vykdoma komanda operuoja (paprastai nurodomi bylą, kompiuterių, vartotojų vardai ir pan.)

## 1s komandos naudojimo su argumentais ir be jų pavyzdys

Panagrinėkime komandą **ls**, kuri skirta katalogo turinio atvaizdavimui.

Įvykdžius komandą **ls be argumentų** pateikiamas failų, esančių einamajame kataloge, sąrašas:

Vykdydami komandas matote jų rezultatus, išvestus į ekraną. Simboliai „>“ bei „>>“ leidžia rezultatus išvesti ne į ekraną, o į norimą failą.

Naudojant „>“ į nurodytą failą duomenys įrašomi ištrinant visą prieš tai buvusią informaciją.

Naudojant „>>“ duomenys pridedami nurodyto failo pabaigoje (t.y. failas papildomas). Abiem šiais atvejais ekrane rezultatų nematysime, gali būti išvedami tik pranešimai apie klaidas, kilusias komandos vykdymo metu.

Vykdamas komandas, galima komandos rezultatus išvesti ne tik į ekraną ar failą, bet taip pat perduoti juos kitai komandai. Tam naudojamas kanalo (angl. pipe) mechanizmas, kuris komandinėje eilutėje nurodomas simboliu |.

## Vykdytų komandų panaudojimas

Formuojant naujas komandas, kartais naudinga pasinaudoti vykdytų komandų sąrašu. Šio sąrašo turinį galima pamatyti komandos **history** pagalba:

```
$ history
```

Pagal nutylėjimą bus išvedama visa vykdytų komandų istorija (pagal nutylėjimą saugoma 500 paskutinių). Jei norima peržiūrėti, pavyzdžiui, tik paskutines 10, galima rašyti:

```
$ history 10
```

Naudojant simbolį **!** (išskyrus tuos atvejus, kai po **!** seka tarpo, tabuliacijos ar naujos eilutės pradžios simboliai) galima iškviesti vykdytų komandų sąrašė esančias komandas:

- **!!** - žymi tai, kad yra kartojama prieš tai vykdyta komanda.
- **!n** - kreipiasi į vykdytų komandų sąrašo n-tą eilutę.
- **!-n** - kreipiasi į eilutę, kurios numeris gaunamas iš einamosios eilutės numerio atėmus n reikšmę.
- **!str** - kreipiasi į paskutinę vykdytą komandą, kuri prasideda seka "str".
- **!?str** - kreipiasi į paskutinę vykdytą komandą, kurioje randa seką "str"

man **puslapiai**

UNIX komandos turi visą eilę privalomų ir neprivalomų parinkčių. Todėl atsiminti kokios parinktys galimos vienai ar kitai komandai praktiškai neįmanoma, o galiausiai to ir nereikia. Šią informaciją galima rasti UNIX pagalbos sistemoje, kuri dar vadinama virtualiu vadovėliu arba **man** puslapiais (angl. **man** pages). **man** puslapiuose pateikiama informacija ne tik apie pagrindines UNIX komandas, tačiau taip pat ir apie C funkcijas, sisteminius kvietinius ir pan. Vadovėlyje esančią informaciją galima peržiūrėti naudojant komandą **man**. Ieškoma informacija apie dominantį objektą yra pateikiama terminalo lange tekstiniu pavidalu.

Bendras naudojimosi **man** komanda aprašas:

```
man [parinktis] dominantis_objektas
```

Norint gauti informaciją apie objektą iš atitinkamo skyriaus naudojama **man** komanda **su parinktimi -s**.

Dalis	Paskirtis
NAME	Aprašoma komandos paskirtis
SYNOPSIS	Pateikiama <a href="#">komandos panaudojimo sintaksė</a>
DESCRIPTION	Pateikiamas išsamus komandos aprašymas
OPTIONS	Pateikiama parinkčių panaudojimo paskirtis
OPERANDS	Pateikiama komandos operandų prasmė
ENVIRONMENT VARIABLES	Pateikiami aplinkos kintamieji, kurių reikšmė turi įtakos komandos vykdymo rezultatui
EXAMPLES	Pateikiami komandos panaudojimo pavyzdžiai
EXIT STATUS	Pateikiamas komandos pabaigos kodo išaiškinimas
ATTRIBUTES	Pateikiama informacija apie komandos atributų reikšmes
SEE ALSO	Pateikiamos komandos, kuriomis patariama pasidomėti vykdant nagrinėjamą komandą
NOTES	Pastabos

**man -k** **ir** apropos

**man -k** ar **apropos** pateikia sąrašą komandų, kurių **pavadinimas** arba **aprašymas** turi komandos argumente nurodytą raktinį žodį. Sąrašas sudarytas iš trumpų komandų aprašymų bei skyriaus pavadinimo, kuriame reiktų ieškoti detalesnės informacijos. Jei išvedamas sąrašas yra per ilgas ir netelpa į terminalo langą, galima panaudojant komandą **more** ir informaciją išvesti puslapiais:

**whatis**

**whatis** komanda pateikia sąrašą komandų, kurių **pavadinimas** turi komandos argumente nurodytą raktinį žodį. **whatis** panaudojimo sintaksė:

**info** **puslapiai**

Ne visa dokumentacija yra pateikiama **man** puslapiuose. Kai kuriais atvejais daugiau informacijos gali būti pateikiama kitame pagalbos vartotojui resurse – t.y. **info** puslapiuose. **info** dokumentus galima peržiūrėti **info** dokumentų peržiūrai skirtoje aplinkoje, iškviečiamoje UNIX komandos **info** pagalba. Bendra **info** panaudojimo sintaksė:

```
info [dominancio_objekto_pavadinimas]
```

Pavyzdžiui, norėdami peržiūrėti informacinį puslapį apie komandą **ls**, turėtume rašyti:

```
info ls
```

Klavišas	Paskirtis
<b>q</b>	išeiti iš info puslapių peržiūros aplinkos
<b>?</b>	trumpas valdymo kombinacijų sąrašas
<b>h</b>	pagalba ("info" sistemos aprašymas)
<b>/zodis_fraze</b>	nurodyto žodžio/frazės paieška
<b>' ' (tarpas)</b>	prasukt tekstą vienu ekranu pirmyn
<b>Del, Backspace, b</b>	prasukt tekstą vienu ekranu atgal
<b>Tab</b>	perstumia kursorių ant sekančios nuorodos
<b>Enter</b>	pereina į mazgą, į kurį rodo nuoroda
<b>rodyklės</b>	kursoriaus valdymas
<b>n</b>	sekanis ("next") mazgas (perėjimui tarp dokumento skyrių)
<b>p</b>	ankstesnis ("previous") mazgas
<b>u</b>	aukščiau esantis ("up") mazgas

## Kelio vardas (*path*)

Failai Unix nurodomi naudojant **absoliučius** arba **santykinius** kelio vardus. Absoliutus kelio vardas yra nurodomas nuo failų sistemos šaknies, pavyzdžiui, absoliutus kelio vardas **/home/agne/tekstas** žymėtų failą **tekstas**, esantį kataloge **agne**, kuris randasi **home** kataloge. Tokį absoliutų kelio iki failo vardą naudojame tada, kai norime kreiptis į failą, esantį kažkur kitur failų sistemoje, o ne tame kataloge, kuriame dirbame. Kreipimuisi į failus galime naudoti ir santykinius failų kelio vardus, jie nėra pradedami **/** ženklu, o yra nurodomi einamojo katalogo atžvilgiu. Pavyzdžiui, Agnė, būdama savo namų kataloge **/home/agne** gali kreiptis į failą **tekstas** tiesiog nurodydama šio failo vardą, tačiau jei ji yra kataloge **/home**, tai nurodydama failą **tekstas** gali naudoti santykinį kelią **agne/tekstas** arba **./agne/tekstas**. Čia **.** žymi einamąjį katalogą.

Specialūs katalogų vardai:

- `/` - šakninis katalogas arba katalogų skyriklis
- `.` - einamasis katalogas
- `..` - vienu hierarchijos lygiu aukščiau failų sistemoje esantis katalogas

## Darbinis katalogas

---

### Tipiniai katalogai ir jų paskirtis

- **`/bin`, `/usr/bin`, `/usr/local/bin`** – kataloguose saugomi komandų vykdomieji failai (komandas atitinkančios programos).
- **`/dev`** – specialūs įrenginių failai.
- **`/etc`** – sistemos konfigūraciją aprašantys failai.
- **`/home`** – vartotojų namų katalogai. Šiame kataloge kiekvienas sistemos vartotojas turi savo katalogą, kuriame laiko savus failus.
- **`/lib`, `/usr/lib`, `/usr/local/lib`** – dinaminės bibliotekos.
- **`/opt`** – stambūs programiniai paketai (vieno paketo failai saugomi viename kataloge ir giliau, o ne išskirstyti po kitus katalogus).
- **`/proc`** – informacija apie Unix sistemoje paleistas programas.
- **`/sbin`, `/usr/sbin`** – sisteminių komandų vykdomieji failai. Čia yra saugomos programos, kurios yra naudojamos sistemai administruoti.
- **`/tmp`** – laikini failai. Tai katalogas laikiniams failams saugoti. Periodiškai trunami (pvz.: perstartuojant OS visi čia saugomi failai ištrinami).
- **`/usr`** – taikomosios programos, jų duomenys, dokumentacija ir t.t.
- **`/var`** – kintantys duomenys (sistemos žurnalai, duomenų bazės ir pan.).

Failai gali turėti **išplėtimus** atskirtus ".", pvz.: **testas.txt**. OS failų išplėtimų niekaip nenaudoja. Išplėtimus gali naudoti taikomosios programos.

### pwd - darbinio katalogo sužinojimas

Komanda **pwd** išveda absoliutų kelią iki einamojo katalogo.

### cd - darbinio katalogo keitimas

Komanda **cd** skirta darbinio katalogo keitimui (perėjimui iš vieno katalogo į kitą). Operandu nurodomas katalogas kuris sėkmingai įvykdžius komandą taps darbiniu (į kurį norima pereiti). Jis gali būti nurodomas naudojant absoliutų arba santykinį kelią. Pavyzdžiui:

### ls - katalogo turinio peržiūra

Komanda **ls** skirta katalogo turinio atvaizdavimui. Ji išveda sąrašą failų bei pakatalogių, esančių nurodytame kataloge. Norint identifikuoti failo tipą, galima naudoti parinktį **-F**. Tokiu atveju po katalogų vardų atsiras „/“ simbolis. Kaip identifikuojami kitų tipų failai galima sužinoti pasiskaičius vadovėlyje (**man ls**). Pavyzdžiui:

### find - paieška

Komanda **find** skirta paieškai failų sistemoje. Supaprastinta komandos sintaksė:

```
find [-L|-H] pradiniskelias... išraiška_ko_ieškoti
```

Parinktys:

- **-L** – jei randama simbolinė nuoroda - išvesti informaciją apie jos rodomą objektą, o jei tokio nėra - apie pačią nuorodą;
- **-H** – tas pats, kaip **-L**, bet galioja tik komandinėje eilutėje nurodytiems pradiniais keliams.
- *pradiniskelis* – vienas ar keli (atskirti tarpu) katalogai, kuriuose ieškoti.
- *išraiška\_ko\_ieškoti* – sąlygos, nurodančios ko ir kaip ieškome (pilną išraiškos sintaksės aprašymą galite rasti **man find** arba POSIX standarte).

### stat - detali informacija apie failą

Komanda **stat** nėra standartizuota, tačiau sutinkama daugumoje UNIX sistemų. Ši komanda išveda smulkia informaciją apie jai nurodytus failų sistemos objektus (failus, katalogus, simbolines nuorodas).

### touch - naujo tuščio failo kūrimas arba datų pakeitimas

Komanda **touch** naudojama norint pakeisti failo sukūrimo, paskutinio modifikavimo ar skaitymo datas arba naujo tuščio failo sukūrimui.

```
$ touch failas1
```

Šiuo atveju **failas1** datoms bus priskirtas dabartinis sistemos laikas, o jei tokio failo/katalogo ar simb. nuorodos nėra - bus sukurtas tuščias failas. Komandai **touch** galima nurodyti daugiau vardų - tuomet atitinkami veiksmai bus atlikti su kiekvienu iš jų:

### mkdir - naujo katalogo sukūrimas

Naujo katalogo sukūrimui naudojama komanda **mkdir**, kuriai nurodomas kuriamo katalogo vardas (absoliutus arba santykinis kelias). Pavyzdžiui:

```
$ mkdir antras
```

### rmdir - tuščio katalogo naikinimas

Ištrinti tuščią katalogą galima komanda **rmdir**. Ši komanda trina argumentais nurodytus tuščius katalogus, pvz.:

```
$ rmdir katalogas1
```

### cp - failo kopijavimas

Failus į katalogus galima nukopijuoti naudojant komandą **cp**. Šioje komandoje pirmu operandu nurodoma iš kur ir kokį failą kopijuosime, o antruoju operandu nurodoma vieta, į kurią bus įrašyta failo kopija, kartu galima nurodyti ir naują vardą, tokiu atveju perkeltas failas įgaus naują vardą. Pavyzdžiui:

### mv - failo ar kataogo pervadinimas ir/arba perkėlimas į kitą vietą

Failas perkeliamas į kitą katalogą naudojant **mv** komandą, tik šiuo atveju failo nelieka ten, kur jis buvo. Komandą **mv** galima panaudoti ir norint pakeisti failo vardą.

```
$ mv tekstas1 tekstas2
```



Ši komanda pervadins darbiniame kataloge esantį failą **tekstas1** į **tekstas2**.

Jei komandai **mv** argumentais nurodysim daugiau nei du failus ar katalogus ir paskutinis argumentas bus katalogas – komanda perkels visus nurodytus failus į paskutiniu argumentu nurodytą katalogą, pvz.:

```
$ mv failas1 ../katalogas2 /tmp/failas3 katalogas4
```

## rm - failo ar katalogo trynimas

Failus trina komanda **rm**. Šiai komandai nurodomas vienas arba keli failai, kuriuos norima ištrinti. Pavyzdžiui:

```
$ rm tekstas1
```

## ln - nuorodų kūrimas

Komanda **ln** skirta simbolinių ir kietų nuorodų kūrimui.

Kietos nuorodos kūrimas:

```
$ ln failas1 failas2
```

Ši komanda sukurs einamajame kataloge antrą įrašą **failas2**, rodantį į tą patį failą **failas1**. Abu įrašai lygiaverčiai, t.y. galima ištrinti bet kurį iš jų, o failas bus pasiekiamas likusiu vardu. Dėl šios priežasties kietoms nuorodoms yra apribojimų: kietos nuorodos veikia tik tarp failų (neišeina to pačio katalogo įrašyti į kelis katalogus), be to neišeina sukurti kietos nuorodos į kito vartotojo failą, ar į failą esantį kitoje failų sistemoje.

Kuriant simbolines nuorodas nurodoma komandos **-s** parinktis.

```
$ ln -s duomenys1 nuroda1
```

Darbiniame kataloge bus sukurta simbolinė nuoroda vardu **nuroda1** rodanti į failą ar katalogą **duomenys1** einamajame kataloge. Failo ar katalogo **duomenys1** gali ir nebūti (**ln** komanda jokios klaidos nerodys ir sukurs argumentais nurodytą nuorodą), tačiau jei kokia nors programa bandys naudoti failą į kurį rodo tokia nuoroda – gaus klaidą. Kitaip sakant simbolinė nuoroda su "tikru" failu ar katalogu mažai susijusi.

Jei komandai **ln** nurodoma daugiau nei du argumentai ir paskutinis iš jų yra egzistuojantis katalogas – komanda sukurs simbolines nuorodas į nurodytus failus ar katalogus paskutiniu argumentu nurodytame kataloge, pvz.:

```
$ ln -s failas1 ../katalogas2 /home/agne/failas3 katalogas4
```

Komanda kataloge **katalogas4** sukurs simbolines nuorodas į pirmais trim argumentais nurodytus failus ar katalogus, t.y. bus sukurtos simbolinės nuorodos: **katalogas4/failas1**, **katalogas4/katalogas2** ir **katalogas4/failas3**, atitinkamai rodančios į **failas1**, **../katalogas2** ir **/home/agne/failas3**.

Tiek kuriant kietas, tiek ir simbolines nuorodas gali būti naudojami ir absoliutūs ir santykiniai keliai.

- Susipažinkime su UNIX komanda **who**, kuri pagal nutylėjimą pateikia informaciją apie prisijungusius prie sistemos vartotojus. Įveskite komandą **who**:

```
$ who
```

- Paanalizuokite gautą informaciją, identifikuodami laukus, kuriuose nurodomi prisijungę prie sistemos vartotojai, terminalo identifikatorius bei paskutinio prisijungimo laikas ir vieta.
- **who** komandos funkcionalumas gali būti keičiamas nurodant parinktis ir operandus. Pavyzdžiui, norėdami išvesti **who** komandos gražinamų rezultatų laukų pavadinimus, galime naudoti parinktį **-H**. Išbandykite **who** be parinkčių ir su parinktimi **-H**:

```
who
```

```
who -H
```

- Bendra komandos **who** argumentų panaudojimo sintaksė (paanalizuokite aprašą):

```
who [-abdHlmpqrstTu] [file | am i | am I]
```

1. Namų kataloge susikurkite katalogą **lab1**

**Mkdir funkcija**

2. Kataloge **lab1** sukurkite naują tuščią failą **tekstas**.

➤ Arba >> failas

3. Pervadinkite failą **tekstas** į **tekstas1**.

**Mv tekstas tekstas1**

4. **lab1** kataloge sukurkite katalogą **pirmas**.

**Mkdir pirmas**

5. Nukopijuokite **tekstas1** į katalogą **pirmas**.

**Cp tekstas1 /pirmas**

6. Išmeskite failą **tekstas1** iš **lab1**.

**Rm lab1/tekstas1**

7. **lab1** kataloge sukurkite katalogą **antras**.

**Mkdir antras**

8. Failą **tekstas1** nukopijuokite iš katalogo **pirmas** į katalogą **antras** vardu **tekstas2**.

**Cp pirmas/tekstas1 antras/tekstas2**

9. Pakeiskite failo **tekstas2** vardą į **tekstas3**.

Mv tekstas2 tekstas3

10. Sukurkite kietą nuorodą **tekstas4** kataloge **antras** į failą **tekstas1** (palyginkite abiejų failų i-node numerius).

Ln tekstas1 antras/tekstas4

11. Sukurkite kataloge **antras** katalogą **trecias**.

Mkdir antras/trecias

12. kataloge **trecias** sukurkite simbolines nuorodas į failus **tekstas1** ir **tekstas4** bei katalogą **lab1** (patikrinkite į ką rodo simbolinės nuorodos).

Ln -s pirmas/tekstas1 antras/tekstas4 lab1

13. Ištrinkite katalogą **pirmas** su visu jo turiniu.

Rm -F pirmas

14. Patikrinkite ar pasikeitė **tekstas4** i-node numeris ir nuorodų (*links*) skaičius.

Ls -l tekstas4

L1/2

more

Komanda **more** skirta failo turinio išvedimui puslapiais. Bendras komandos aprašas:

```
more [-cdfllrsuw] [-lines] [+ linenum] [+/- pattern] [file]...
```

Informacija yra išvedama į virtualaus terminalo langą telpančiomis failo turinio porcijomis (toliau tekste naudojamas terminas **puslapis**). Spaudžiant tarpo klavišą išvedamas sekantis puslapis. Spaudžiant „Enter“ klavišą informacija papildoma viena papildomai išvesta eilute. Spaudžiant „q“ klavišą išvedimas stabdomas.

Parinktys (ne visos):

- **-lines** - nurodomas norimas eilučių puslapyje kiekis.
- **+linenum** - nurodoma, nuo kurios failo eilutės išvesti failo turinį puslapiais.
- **+/-pattern** - nurodoma, kad prieš išvedant failo turinį būtų ieškoma eilutė, turinti nurodytą seką „pattern“.

## head

Komanda **head** skirta pirmųjų keletos failo eilučių išvedimui. Jei failas nenurodytas, tuomet išvedamos iš klaviatūros įvestos eilutės. Bendras komandos aprašas:

```
head [-number | -n number] [filename]...
```

Parinktys:

- **-number** arba **-n number** - nurodomas pirmųjų išvedamų eilučių skaičius. Pagal nutylėjimą, išvedamų eilučių skaičius yra 10.
- **filename1 filename2 ...** - komandos operanduose galima nurodyti daugiau nei vieną failą. Tuomet komanda išves standartinį arba **-number** parinktimi nurodytą eilučių skaičių kiekvienam operanduose nurodytam failui, failus atskiriant "==" Failo vardas <==" skyrikliu.

## tail

Komanda **tail** skirta paskutinių failo eilučių išvedimui. **tail** kopijuoja nurodyto failo eilutes į terminalo langą, pradedant nustatyta vieta iki failo pabaigos. Jei failas nenurodytas, išvedama tai, kas yra įvedama klaviatūra. Bendras komandos aprašas:

```
tail [-f|-r] [-c number | -n number] [file]

tail [+/-[number][lbc][f]] [file]

tail [+/-[number][l][r|f]] [file]
```

Failo turinio kopijavimas prasideda nuo pozicijos atstumu **+number** nuo failo pradžios arba **-number** nuo failo galo. Jeigu **number** reikšmė nenurodyta, tai priskiriama reikšmė lygi -10. Turi būti nurodytas tik vienas iš ženklų **+** arba **-**. Atstumas užduodamas eilutėmis, blokais arba simboliais atsižvelgiant į tai, kokia parinktis naudojama po skaičiaus - **l**, **b** arba **c**. Jeigu matavimo vienetas neužduotas, tai atstumas matuojamas eilutėmis. Negalima nurodyti daugiau vieno matavimo vieneto. Alternatyviai atstumą eilutėmis arba simboliais galima nurodyti naudojant ir parinktis atitinkamai **-c number** (nurodo baitų kiekį), bei **-n number** (nurodo eilučių kiekį).

Esant parinkčiai **-f**, **tail** komanda, išvedusi failo turinį iki pabaigos, nepasibaigia, o laukia kol į failą bus įrašyta daugiau duomenų ir juos išveda. Toks režimas naudingas norint stebėti failo turinį, kai į jį rašo kiti procesai.

## cat

Komanda **cat** skirta trumpų failų peržiūrai. Jei **cat** komandoje failas, kurį norima skaityti ir jo turinį išvesti į terminalo langą, nenurodomas - tuomet duomenys imami iš klaviatūros. Bendras komandos aprašas:

```
cat [-nbsuvet] [file...]
```

## Nano redaktorius

Tai vienas iš paprastesnių teksto redaktorių. Dirbant šiuo redaktoriumi pelė nenaudojama. Dauguma **nano** redaktoriaus komandų yra iškviečiamos laikant nuspaustą **<Ctrl>** klavišą ir renkant reikiamą raidę. Šiame tekste (o ir pačiame redaktoriuje taip pat) **<Ctrl>** klavišas vaizduojamas **"^"** simbolio pagalba. Pavyzdžiui, **^X** reiškia, kad nuspaudę **<Ctrl>** klavišą turime spausti **"X"** raidę.

Dalis **nano** redaktoriaus komandų yra iškviečiamos spaudžiant ne **<Ctrl>**, bet **<Esc>**, **<Alt>**, **<Ctrl-Alt>** ar kitą (priklausomai nuo klaviatūros nustatymų) klavišą. Tokie klavišai vaizduojami "M-" simboliu.

### *nano redaktoriaus iškvietimas*

Redaktorius iškviečiamas **nano** komandos pagalba:

```
$ nano
```

Klavišų kombinacija	Paskirtis
<b>^A</b>	Perkelti kursorių į eilutės pradžią
<b>^E</b>	Perkelti kursorių į eilutės pabaigą
<b>^Y</b>	Grįžti į prieš tai buvusį puslapį
<b>^V</b>	Pereiti prie sekančio puslapio
<b>^_</b>	Perkelti kursorių į tam tikrą norimą vietą (Būsenos eilutėje nurodant norimą eilutę ir stulpelį).
<b>M-/ arba ^_-^V</b>	perkelia kursorių į failo pabaigą
<b>M-\ arba ^_-^Y</b>	perkelia kursorių į failo pradžią
<b>^C</b>	Būsenos eilutėje išvesti informaciją apie tai, kurioje eilutėje/stulpelyje yra kursoriai.
<b>^W</b>	Teksto paieška atidarytame faile (būsenos eilutėje nurodome raktinį paieškos žodį). Paieška vykdoma nuo einamosios kursoriaus pozicijos
<b>M-&gt;</b>	Pereiti prie sekančio atidaryto failo
<b>M-&lt;</b>	Grįžti prie ankstesnio atidaryto failo

Redagavimui skirti klavišai:

Klavišų kombinacija	Paskirtis
simbolis	Simbolio įterpimas vykdomas einamojoje kursoriaus pozicijoje rašant norimą simbolį.
<b>^D</b>	trinamas simbolis (einamojoje kursoriaus pozicijoje)
<b>&lt;BackSpace&gt;</b>	trinamas simbolis (esantis prieš kursorių)

^K	Trinama eilutė (einamojoje kursoriaus pozicijoje)
^\	Simbolio/sekos suradimas ir pakeitimas norimu simboliu/seka (nuo einamosios kursoriaus pozicijos)

ailo saugojimui skirti klavišai:

Klavišų kombinacija	Paskirtis
^O	Failo turinio išsaugojimas neuždarant failo ir neišeinant iš nano redaktoriaus
^X	Darbo su nano užbaigimas (jei faile buvo pakeitimų, redaktorius užklauso, ar norite tuos pakeitimus išsaugoti)

## grep

```
grep [-iv] sablonas [failas]
```

- **-i** – neskirti didžiųjų ir mažųjų raidžių ('a' atitinka ir 'a' ir 'A' simbolius);
- **-v** – invertuoti rezultatą (išvesti eilutes neatitinkančias šablono).

Komanda **grep** - tai filtras atrenkantis šabloną (reguliarią išraišką) tenkinančias teksto eilutes. Kitaip sakant **grep** - tai programa, kuri skaito po vieną eilutę, tikrina ar toje eilutėje yra argumentu nurodyta seka ir jei tokia seka yra - eilutė išvedama. Jei failas nenurodytas - analizuojamas įvedamas tekstas (pvz.: klaviatūra vedamos eilutės iki ^D paspaudimo).

## file

```
file [-hi] failas1 ...
```

Komanda **file** skirta nustatyti failo tipui. Vykiant komandą **file** atliekama bandymų serija su kiekvienu komandos operanduose pateiktu failu bandant jį klasifikuoti. Jei failas yra tekstinis, **file** komanda bando atpažinti turinio struktūrą (programavimo kalbą, kodavimą ir t.t.). Netuščias failas, kurio turinys neatpažįstamas, identifikuojamas kaip **data** – duomenys. Failo turinys gali būti identifikuojamas klaidingai, nes atliekami bandymai nėra labai patikimi. Parinktys (ne visos):

- **-h** – jei nurodyta simbolinė nuoroda, identifikuoti, kaip simbolinę nuorodą, neanalizuojant failo, į kurį ji rodo.
- **-i** – jei nurodytas paprastas failas, neanalizuoti jo turinio (nebandyti identifikuoti pagal turinį).

## strings

Išveda teksto fragmentus, rastus nurodytame faile (pavyzdžiui, vykdomajame faile).

Komanda **strings** ieško **ASCII** sekų dvejetainiame faile, kai seką sudaro 4 (pagal nutylėjimą) ar daugiau spausdinamų simbolių, po kurių seka nespausdinamas simbolis.

Veikimas pagal nutylėjimą priklauso nuo OS, 193.219.36.233 mašinoje **strings** pagal nutylėjimą analizuoja visą failą.

Parinktys:

- **-d** – analizuojama tik vykdomojo failo dalis, kuri bus įkelta į RAM šią programą vykdant (t.y. kai kurios vykdomojo failo dalys gali būti ignoruojamos);
- **-n N** – galima pakeisti pagal nutylėjimą naudojamą minimalų simbolių eilutės ilgį (čia **N** – sveikas skaičius);
- **-t formatas** – prieš kiekvieną surastą seką spausdina sekos poslinkį baitais nuo failo pradžios. Formatas gali būti:
  - **d** – poslinkis nurodomas dešimtainėj sistemoj,
  - **o** – poslinkis nurodomas aštuntainėj sistemoj,
  - **x** – poslinkis nurodomas šešioliktainėj sistemoj.

## diff

**diff** komanda išveda skirtumus tarp dviejų argumentais nurodytų failų, kitaip sakant išveda sąrašą pakeitimų, kurių reikia tam, kad **failas1** sutaptų su failu **failas2**. Jei failai yra identiški **diff** nieko neišveda. Standartinė **diff** naudojimo sintaksė:

```
diff [-c | -e | -f | -u | -C n | -U n] [-br] failas1 failas2
```

Parinktys:

- **-c, -e, -f, -u** – nurodo išvedimo formatą
- **-C n, -U n** – naudojamas tas pats formatas, kaip **-c** ar **-u**, bet išvedama ne 3, o **n** (sveikas skaičius) konteksto eilučių.
- **-b** – lyginant failus ignoruojami tarpo, tabuliacijos ir analogiški simboliai eilutės gale
- **-r** – jei **failas1** ir **failas2** – katalogai, jų turinys lyginamas rekursiškai (t.y. lyginami failai vienodais pavadinimais esantys nurodytuose kataloguose)

Jei abu nurodyti failai yra katalogai, **diff** lygina failus vienodais vardais esančius abiejuose kataloguose. Jei nenurodyta **-r**, šis palyginimas nėra rekursinis. Jei vienas iš nurodytų failų yra katalogas, o kitas yra paprastas failas, **diff** lygina nurodytą failą su kataloge esančiu tokio paties vardo failu (naudojamas tik failo vardas, be kelio).

Rezultatų formatas pagal nutylėjimą (**n1**, **n2** atitinka failo **failas1** eilučių numerius ir **n3**, **n4** failo **failas2** eilučių numerius) :

- **n1an3,n4** – reiškia, kad po **failas1** esančios eilutės **n1** reikia pridėti (**a** – *add*) eilutes nuo **n3** iki **n4**, norint gauti **failas2**.
- **n1,n2dn3** – reiškia, kad iš **failas1** išmetus (**d** – *delete*) eilutes nuo **n1** iki **n2** bus gautas **failas2** (t.y. **failas2** trūksta eilučių po **n3**);
- **n1,n2cn3,n4** – reiškia, kad vietoj (**c** – *change*) **failas1** eilučių nuo **n1** iki **n2**, **failas2** yra eilutės nuo **n3** iki **n4**;  
Jei **n1** lygus **n2** arba **n3** lygus **n4** – antras skaičius praleidžiamas, pavyzdžiui vietoj **7,7c7,7** lieka **7c7** (t.y. pakeista 7 eilutė).

## uniq

Komanda išfiltruoja pasikartojančias eilutes palikdama po vieną pasikartojančią eilutę išvedime.

```
uniq [-c | -d | -u ] [ -f N ] [-s N] [duomenys [rezultatai]]
```

Parinktys:

- **-c** – prieš kiekvieną išvesties eilutę pateikiamas skaičius, nurodantis kiek kartų ši eilutė kartojasi.
- **-d** – visai neišveda eilučių, kurios nesikartoja.
- **-f N** – ignoruoja pirmus N žodžių kiekvienoje eilutėje (jų neįvertina). Žodis yra simbolių seka be tarpų ir tabuliacijos simbolių.
- **-s N** – ignoruoja pirmus N simbolių. Jei nurodyta kartu su **-f** parinktimi, yra ignoruojami pirmi N simbolių po praleistų žodžių.
- **-u** – visai neišveda eilučių, kurios kartojasi.

## od

```
od [-v] [-A address_base] [-j skip] [-N count] [-t type_string]... [file...]
```

Komanda **od** išveda failo turinį įvairiais formatais (simboliais, aštuntainiu, dešimtainiu, šešioliktainiu, slankaus kablelio). Jei išvedimo formatas nenurodomas, pagal nutylėjimą yra priimama parinktis **-t o2**.

Parinktys:

- **-A address\_base** – adreso išvedimo formatas, *address\_base* – vienas iš simbolių **d** (dešimtainis), **o** (aštuntainis), **x** (šešioliktainis), **n** (adresas neišvedamas);
- **-b** – atvaizduoja failo baitus aštuntainėje sistemoje;
- **-c** – atvaizduoja simbolį, atitinkantį kiekvienam failo baitui. Kai kurie nespausdinami simboliai atvaizduojami C programavimo kalbos formatu ( **\0**, **\b**, **\f**, **\n**, **\r**, **\t** ), kiti **\XXX** formatu (X - aštuntainis skaičius);
- **-d (-t u2)** – atvaizduoja žodžius (2 baitus, nepriklausomai nuo mašinos žodžio ilgio) kaip dešimtainius skaičius be ženklo;
- **-j skip** – praleidžia *skip* (sveikas skaičius) baitų nuo failo pradžios;
- **-N count** – formatuoja ne daugiau kaip *count* (sveikas skaičius) baitų;
- **-o** – Interpretuoja žodžius aštuntainėje sistemoje. Tai yra ekvivalentu parinkčiai **-t o2**;
- **-x** – Interpretuoja žodžius šešioliktainėje sistemoje. Tai yra ekvivalentu parinkčiai **-t x2.**;
- **-v** – išveda visus duomenis (be šios parinkties vienodas išvedimas nekartojamas vietoj tokių eilučių išvedant \*);
- **-t type\_string** – nurodo išvedimo formatą, *type\_string* – eilutė sudaryta iš išvedimo formatą nusakančių simbolių **a** („įvardintas“ simbolis) , **c** (simbolis) , **d** (sveikas skaičius su ženklu) , **f** (slankaus taško skaičius) , **o** (aštuntainis skaičius), **u** (dešimtainis skaičius be ženklo), **x** (šešioliktainis skaičius). Po kiekvieno iš **dfoux** simbolių gali eiti sveikas skaičius, nurodantis tipo naudojamą baitų skaičių. Vietoje skaičiaus po **f** gali eiti vienas iš **FDL** simbolių (*float*, *double*, *long double* C kalbos tipai), o po **doux** gali eiti vienas iš **CSIL** simbolių (*char*, *short*, *int*, *long* C kalbos tipai). Jei *type\_string* nurodyti keli formatai (daugiau nei vienas tipas) – duomenys išvedami kiekvienu nurodytu formatu.

## wc

```
wc [-c|-m] [-lw] [file...]
```

Komanda **wc** parodo faile esančių eilučių, žodžių ir simbolių kiekį.

Parinktys:

- **-c** – išveda baitų kiekį faile;
- **-m** – išveda simbolių kiekį faile (simbolį gali sudaryti daugiau nei 1 baitas, "multi-byte" simboliai);
- **-l** – išveda eilučių kiekį faile;
- **-w** – išveda žodžių kiekį (žodžius skiria tarpo, tabuliacijos, naujos eilutės ar analogiški simboliai).



## iconv

```
iconv [-cs] -f FromCode -t ToCode [ FileName... ]
```

Komanda **iconv** skirta failo simbolių ar jų sekų konvertavimui iš vieno kodavimo į kitą. Duomenų kodavimas identifikuojamas parametru *FromCode*, o rezultato *ToCode*. Jei duomenų failo vardas nenurodomas komandų eilutėje, **iconv** komanda skaito duomenis iš standartinės įvesties (konsolės).

Parinktys:

- **-c** – praleisti (neišvedinėti) simbolių, kurie negali būti konvertuoti;
- **-f FromCode** – nurodo duomenų kodavimą;
- **-l** – išveda palaikomų kodavimų sąrašą (galimas *FromCode* ir *ToCode* parametrų reikšmės);
- **-s** – nespausdina klaidų, kilusių dėl nekonveruojamų simbolių, pranešimų;
- **-t ToCode** – nurodo rezultatų kodavimą.

Nacionaliniai (ne angliški) simboliai gali būti koduojami labai įvairiai, tai priklauso nuo naudojamos kodų lentelės (kodavimo). Pavyzdžiui lietuviško teksto kodavimui dažniausiai naudojamos *windows-1257*, *iso-8859-4*, *iso-8859-13*, *utf-8* kodų lentelės. Klientinės programos (email klientai, *www* naršyklės ir kt.) paprastai šiuos kodavimo skirtumus paslepia nuo naudotojo. Tais atvejais, kai klientinė programa neteisingai atpažįsta arba nepalaiko duomenų kodavimo, vienas iš galimų sprendimų - perkoduoti duomenis į programai suprantamą formatą naudojant **iconv** ar analogišką pagalbinę priemonę.

## sort

```
sort [-m] [-o rezfailas] [-bdfinru] [-t C] [-k raktas]... [file...]
```

Komanda **sort** rūšiuoja nurodyto failo eilutes (jei failas nenurodytas – standartinio įvedimo eilutes). Rūšiavimo tvarka nurodoma komandos **sort** parinktimis.

Parinktys:

- **-m** – tik apjungimas (duomenų failas turi būti jau surūšiuotas)
- **-o rezfailas** – surūšiuoti duomenys rašomi į **rezfailas** failą;
- **-b** – ignoruoti tarpus ir tabuliacijos simbolius pradžioje;
- **-d** – vertinti tik tarpus, raides ir skaitmenis;
- **-f** – rūšiuojant mažąsias raides palyginimui paversti didžiosiomis;
- **-i** – ignoruoti nespausdinamus simbolius;
- **-n** – rūšiuoti, kaip skaičius;
- **-r** – rūšiuoti atvirkščia tvarka;
- **-u** – palikti tik vieną eilutę, jei yra kelios, kurios pagal nurodytus parametrus yra lygios;
- **-t C** – naudoti nurodytą simbolį *C*, kaip laukų skirtuką;
- **-k raktas** – nurodo eilutės dalį, pagal kurią rūšiuojama. Parinktis gali būti kartojama, nurodant iki 9 raktų.

Rūšiavimo rakto formatas: pradžia[T][,pabaiga[T]],  
čia pradžia ir pabaiga formatas laukas[.simbolis], o *T* – vienas iš simbolių *bdfinr*, veikiantis, kaip atitinkama **sort** parinktis. Pavyzdžiui:

- **2** – rūšiuoti pagal eilutės dalį pradedant nuo antro lauko iki eilutės pabaigos;
- **2,2** – rūšiuoti pagal antrą lauką;
- **2b,2b** – rūšiuoti pagal antrą lauką nevertinant tarpų;
- **3.2,3.2** – rūšiuoti pagal trečio lauko antrą simbolį (pirmas simbolis – skyriklis).

## split

```
split [-l line_count] [-a suffix_length] [file [name]]  
split -b n[k|m] [-a suffix_length] [file [name]]
```

Komanda **split** sudalija failą į dalis. Dalijimas priklauso nuo nurodytų parinkčių.

Parinktys:

- **-b N** – failas sudalijamas į dalis po N baitų.
- **-b Nk** – failas sudalijamas į dalis po N\*1024 baitų (N kilobaitų).
- **-b Nm** – failas sudalijamas į dalis po N\*1048576 baitų (N megabaitų).
- **-l L** – failas yra sudalijamas į dalis po L eilučių kiekvienoje dalyje.
- Pavyzdžiui:

```
$ split -b 22 newfile.txt new
```

- Ši komanda sudalys failą **newfile.txt** į atskirus failus, kurių vardai bus **newaa**, **newab**, **newac**, ... Kiekvieno šių failų dydis bus 22 baitai.

```
$ split -l 300 file.txt new
```

## cmp

```
cmp [-l | -s] file1 file2
```

Komanda **cmp** palygina du failus **file1** ir **file2**. Komanda **cmp** nieko neišveda, jei failai yra vienodi. Jei failai skiriasi, išvedamas pirmo skirtingo baito bei eilutės numeris, kur skirtumas surastas. Baitai ir eilutės numeruojami nuo 1.

Parinktys:

- **-l** – kiekvienam skirtumui išvesti dešimtainį baito numerį ir besiskiriančius baitus (aštuntainėje sistemoje);
- **-s** – jei failai skiriasi nieko neišvesti, tik grąžinti ne nulinį pabaigos statusą.

Pavyzdžiui:

```
$ cmp file1 file2
```

## cksum

```
cksum [failas...]
```

Komanda **cksum** suskaičiuoja ir išveda failo **failas** kontrolinę sumą bei dydį.

Dažniausiai kontrolinė suma naudojama įsitikinimui, kad failas nėra modifikuotas (be klaidų persiųstas, neredaguotas, neužkrėstas virusais ir t.t.). Tuo pačiu tikslu dažniau naudojamos MD5 ar SHA kontrolinės sumos skaičiuojančios utilitos, tačiau jos, skirtingai nei **cksum** nėra standartinės.

*more*

---

- Išveskite **/usr/sbin** katalogo turinį (failų sąrašą) į failą **sbin.txt**

Ls -l /usr/sbin > sbin.txt

- Peržiūrėkite failą **sbin.txt** puslapiais.

More sbin.txt

- Panaudodami **more** komandą (su reikalingomis parinktimis) atverskite failo **sbin.txt** puslapį nuo 3 įrašo.

More -3 sbin.txt

- Panaudodami **more** komandą (su reikalingomis parinktimis) atverskite failo **sbin.txt** puslapį su įrašais, turinčiais seką "dump".

More +/dump sbin.txt

- *[Advanced]* Peržiūredami failą **sbin.txt** parinkite tokias parinktis, kad failą galėtume peržiūrinėti:
  - po vieną įrašą;
  - po tris įrašus.

L1/3

Kiekvienas UNIX sistemos naudotojas identifikuojamas numeriu (UID, sveiku skaičiumi pradedant nuo 0). Naudotojas, kurio numeris 0 - administratorius (arba **root**'as), šiam naudotojui negalioja dauguma saugumo sistemos apribojimų. Vartotojų hierarchija realizuojama **grupių** pagalba. Kiekviena grupė taip pat identifikuojama sveiku skaičiumi (GID, pradedant nuo 0). Kiekvienas naudotojas turi savo pagrindinę (*primary*) grupę, bet gali priklausyti ir kitoms grupėms.

Todėl UNIX OS mato trijų tipų naudotojus, bandančius naudoti failą:

1. savininkas (**user**, *owner*)

2. grupės nariai (**group**)
3. visi kiti (**other, world**)

Kiekvienam šių tipų gali būti suteikti (arba nesuteikti) leidimai:

1. skaityti (**read**)
2. rašyti (**write**)
3. vykdyti (**execute**)

*PASTABA: norint įeiti į katalogą reikia **execute** teisės, o norint perskaityti jo turinį – **read** teisės, t.y. jei yra tik viena iš šių teisių, galite įeiti į katalogą, bet nematyti jo turinio, arba matyti turinį, bet negalėti įeiti.*

1. **set-user-ID (suid)** – paleidus failą (programą) vykdymui, jis veikia failo savininko, o ne jį paleidusio naudotojo teisėmis;
2. **set-group-ID (sgid)** – paleidus failą vykdymui jis veikia failo grupės teisėmis, o ne iškviečiančio naudotojo grupės teisėmis; taip pat šis bitas įtakoja naujai kuriamų katalogų grupę ir leidimus; yra specifinių savybių skirtingose OS;
3. **sticky bit** – kataloguose, kuriems nustatytas šis leidimas, naudotojai negali trinti kitų naudotojų failų/katalogų (pvz.: /tmp katalogas, į kurį gali rašyti visi naudotojai, bet negali trinti ne savo duomenų); yra specifinių savybių skirtingose OS.

bitas	dvejetainis	aštuntainis	simbolinis	vardas	prasmė
0	0000000000001	00001	----- x	S_IXOTH	leidimas vykdyti kitiems
1	0000000000010	00002	----- w-	S_IWOTH	leidimas rašyti kitiems
2	000000000100	00004	-----r- -	S_IROTH	leidimas skaityti kitiems
3	000000001000	00010	-----x-- -	S_IXGRP	leidimas vykdyti grupei
4	000000010000	00020	-----w--- -	S_IWGRP	leidimas rašyti grupei
5	000000100000	00040	-----r--- -	S_IRGRP	leidimas skaityti grupei
6	000001000000	00100	---x----- -	S_IXUSR	leidimas vykdyti savininkui
7	000010000000	00200	--w----- -	S_IWUSR	leidimas rašyti savininkui
8	000100000000	00400	-r----- -	S_IRUSR	leidimas skaityti savininkui

9	001000000000	01000	----- T	S_ISVTX	<i>sticky bit</i>
10	010000000000	02000	-----S-- -	S_ISGID	<i>set group ID</i>
11	100000000000	04000	---S----- -	S_ISUID	<i>set user ID</i>
	001000000001	01001	----- t		nustatyti 2 bitai: <i>sticky bit</i> ir vykdymas kitiems
	010000001000	02010	-----S-- -		nustatyti 2 bitai: <i>set group ID</i> ir vykdymas grupei
	100001000000	04100	---S----- -		nustatyti 2 bitai: <i>set user ID</i> ir vykdymas savininkui

## Paslėpti failai

Jei failo vardo pirmas simbolis yra taškas ( . ) tai yra paslėptas failas/katalogas. Vienintelis tokio failo skirtumas, kad dalis programų nenaudojant papildomų parinkčių ar panašių priemonių jų nerodo/nenaudoja. Dažniausiai tokie failai naudojami programų nustatymams saugoti.

UNIX failų sistemos kataloguose saugomi įrašai:

1. [paprasti failai](#) (tame tarpe ir kietos nuorodos - t.y. įrašas apie tą patį failą)
2. [katalogai](#)
3. [simbolinės nuorodos](#)
4. [simboliniai įrenginių/specialūs failai](#) – prie sistemos prijungtų, joje esančių fizinių ar virtualių įrenginių abstrakcija (pelė, printeris, klaviatūra, audio ir video kontrolieriai, atsitiktinių skaičių generatorius, ir pan.) naudojama I/O veiksmams su įrenginiais. Rašant į ar skaitan iš tokio failo – duomenys rašomi/skaitomi į/iš atitinkamo įrenginio.
5. [blokiniai įrenginių/specialūs failai](#) – panašūs į simbolinius įrenginių failus, bet dėl atitinkamų įrenginių specifikos informacijos mainai su jais galimi tik fiksuoto dydžio blokais (pvz.: kieti diskai, CD/DVD, flash'ai ir pan.)
6. [įvardyti vamzdžiai](#) – objektai komunikacijai tarp procesų (paleistų programų) ([smulkiau](#)).
7. [soketai](#) – objektai komunikacijai tarp procesų ([smulkiau](#)).

## id – naudotojo identifikatorių nustatymas

Komanda **id** parodo naudotojo UID, GID ir grupių, kurioms priklauso naudotojas sąrašą. Ši komanda gali pateikti informaciją tiek apie vartotoją, kuris iškviėtė šią komandą, tiek apie vartotoją, kurio prisijungimo vardas nurodytas argumentu. Pavyzdžiui:

```
$ id
uid=60017(nijsara) gid=100(users)
```

Laukas "uid" rodo vartotojo identifikacinį numerį sistemoje, skliaustuose rodomas vartotojo prisijungimo vardas, o laukas "gid" rodo pagrindinės grupės, kuriai priklauso vartotojas, numerį bei grupės vardą.

```
$ id vytbol
uid=60377(vytbol) gid=100(users)
```

## groups – grupių, kurioms priklauso naudotojas sąrašas

<!-- neaktualu, kol studentai tik vienoje grupėje -->

*PASTABA: komanda nestandartinė, bet dažnai yra Linux/Solaris/BSD OS.*

Komanda **groups** išveda grupių, kurioms priklauso naudotojas, sąrašą.

```
$ groups
users dest
```

## getent – OS duomenų bazių turinio peržiūra

*PASTABA: komanda nestandartinė, bet dažnai yra Linux/Solaris/BSD OS.*

```
getent database [key...]
```

Argumentai:

- **database** – duomenų bazė, galimos reikšmės: **passwd**, **group**, **hosts**, **services**, **protocols** ...
- **key** – paieškos raktas (jei nurodytas raktas, išvedama ne visa duomenų bazė, o tik raktą atitinkantis įrašas).

## ls – katalogo turinio peržiūra (detaliau)

### *Papildoma informacija*

Komanda **ls** skirta informacijos išvedimui apie failus.

```
ls [-ACFRSacdfigiklmnopqrstux1] [-H|-L] [failas...]
```

Parinktys (ne visos, likusias galite rasti POSIX standarte arba man puslapyje):

- **-A** – rodyti ir paslėptus failus (pirmas simbolis „.“), bet nerodyti „.“ ir „..“ įrašų
- **-F** – nesekti operandais nurodytų simbolių nuorodų, po kiekvieno katalogo rodyti „/“ simbolį, po kiekvieno vykdomojo failo „\*“, po kiekvieno kanalo „|“, po kiekvienos simbolinės nuorodos „@“
- **-R** – rekursiškai rodyti katalogų turinius
- **-S** – rūšiuoti pagal dydį mažėjimo tvarka
- **-a** – rodyti visus failus (kaip **-A**, bet papildomai rodyti „.“ ir „..“)
- **-i** – rodyti failų *i-node* numerius
- **-d** – rodyti informaciją apie operandais nurodytus katalogus ar simbolines nuorodas į katalogus (ne jų turinį)
- **-l** – ilgas išvedimo formatas (apie kiekvieną failą išvedama detali informacija)
- **-p** – po kiekvieno katalogo vardo rašyti „/“
- **-r** – rūšiuoti priešinga tvarka (pagal ką rūšiuojama priklauso nuo kitų parinkčių)
- **-s** – rodyti failo užimamą vietą blokais (bloko dydis priklauso nuo OS)
- **-t** – rūšiuoti pagal modifikavimo laiką

- **-l** – (vienetas) rezultatus išvesti vienu stulpeliu (kai nenaudojama **-l** parinktis)

Igo išvedimo formatas (**-l** parinktis):

```
[S] Tuuugggooo L savininkas grupė NNN data vardas
```

Čia:

- **[S]** – blokų skaičius arba *i-node* numeris (išvedamas jei nurodyta **-s** arba **-i** parinktis)
- **T** – failo tipas
  - - – paprastas failas
  - d – katalogas
  - l – simbolinė nuoroda
  - b – blokinio įrenginio failas
  - c – simbolinio įrenginio failas
  - p – kanalas
  - s – soketas
- uuu, ggg, ooo – savininko, grupės ir kitų leidimai simboliu formatu
- L – failui: kietų nuorodų skaičius, katalogui: jame esančių įrašų apie katalogus skaičius (kiek kataloge yra katalogų, įskaitant „.“ ir „..“)
- savininkas – failo savininko vardas
- grupė – failo grupės vardas
- NNN – failo dydis baitais
  - įrenginių failui vietoj dydžio gali būti išvedama su įrenginiu susijus informacija (paprastai du skaičiai **M**, **m**, kur **M** (*major*) – draiverio/tipo identifikatorius, **m** (*minor*) – tokio tipo įrenginio identifikatorius)
- data – paskutinio modifikavimo data
- vardas – failo [kelio] vardas
  - simbolinei nuorodai: **vardas** -> **adresatas**

## chmod – leidimų keitimas

Prieigos prie failo leidimai keičiami arba nustatomi, naudojant komandą **chmod**.

```
chmod [-R] mode file...
```

Parinktys:

- **-R** – leidimus keisti rekursyviai, t.y. keičiami ir visų nurodytuose kataloguose esančių failų leidimai
- **mode** – nauja leidimų reikšmė, gali būti nurodoma aštuntainiu skaičiumi arba simboliu išraiška.
- Supaprastinta *mode* simbolinės išraiškos struktūra (pilną aprašymą žiūrėkite *POSIX* standarte arba man puslapiuose):

```
[ugoa]+|-|=[rwxXstugo][,...]
```

Čia:

- **ugoa** (pradžioje) – kieno leidimams išraiška taikoma: **u** - savininko, **g** - grupės, **o** - kitų, **a** - visų (tas pats, kaip **ugo**)
- **+|-|=** – kas bus daroma su leidimais: **+** - pridedami nurodyti papildomi, **-** - nuimami nurodyti esami, **=** - nustatomi nurodyti (tie, kurie nenurodyti - nuimami)
- **rwxXstugo** – leidimai: **r**, **w**, **x** - skaitymas/rašymas/vykdydas, **s** - SUID/SGID (priklausomai, kam taikoma), **t** - *sticky-bit*, **X** - vykdydas katalogams, **u**, **g**, **o** - kopijuojami savininko/grupės/kitų leidimai.

## umask – naujai kuriamų failų leidimai

Komanda **umask** nustato, kokie leidimų bitai turi būti išjungiami naujai kuriamiems failams/katalogams.

```
umask [-S] [mask]
```

Parinktys:

- **-S** – *umask* reikšmę išvesti simboliškai išraiška.
- **mask** – nauja *umask* reikšmė, gali būti nurodoma tiek skaičiais, tiek ir simboliais (formatas panašus į **chmod**).
- Naudojimo pavyzdžiai:

- `$ umask`
- `$ umask -S`
- `$ umask 0077`
- `$ umask u=rwx,g=rx,o=`
- `$ umask o+rx`

## du – užimama vieta

Komanda **du** skirta peržiūrėti, kiek vietos failų sistemoje užima nurodytas failas ar katalogas (visi jame esantys failai). Pagal nutylėjimą, POSIX standarte failo dydis yra nusakomas 512 baitų blokais, tačiau Linux **du** komanda naudoja 1024 baitų blokus.

```
du [-a|-s] [-kx] [-H|-L] [file...]
```

Parinktys:

- **-a** – rodyti ir visų nurodytuose kataloguose esančių failų užimamą vietą (pagal nutylėjimą hierarchijoje rodomi tik suminiai katalogų dydžiai)
- **-s** – rodyti tik suminį nurodytų katalogų dydį (nerodo atskirai informacijos apie hierarchiją)
- **-k** – dydį skaičiuoti 1K blokais (~~ne 512 baitų blokais~~)
- **-x** – vertinti tik toje pačioje failų sistemoje esančius failus (hierarchijoje neiti į kitas failų sistemas)
- **-H** – jei komandos eilutėje nurodyta simbolinė nuoroda į katalogą - skaičiuoti to katalogo dydį (bet giliau hierarchijoje sutikus simb. nuorodas pagal jas neiti)
- **-L** – jei komandos eilutėje nurodyta simbolinė nuoroda į katalogą - skaičiuoti to katalogo dydį, taip pat ir pagal hierarchijoje aptiktas simb. nuorodas.

## df – laisva vieta

Komanda **df** parodo laisvos vietos kiekį failų sistemose.

```
df [-k] [-P|-t] [file...]
```

Parinktys:

- **-k** – vietą skaičiuoti 1K blokais (o ne 512 baitų).
- **-P** – alternatyvus išvedimo formatas, su antrašte



PASTABA: *df* skirtingose OS skiriasi (turi specifinių parinkčių)

Jei nurodytas failas (ar keli failai) – *df* rodo laisvą disko vietą kiekį tik tose failų sistemose kurioms priklauso tie failai. Naudojant *df -k* išvedama tokia informacija:

Filesystem	1024-blocks	Used	Available	Capacity	Mounted on
FS įrenginio vardas	FS dydis blokais	Panaudota blokų	Laisvų blokų	Užimta %	Kur primontuota

L1/4

## Procesų medis

**Procesas** – paleistos programos egzempliorius, t.y. jei tą pačią programą paleisime antrą kartą, kol pirmoji dar nepasibaigė - turėsime du procesus, vykdančius tą pačią programą.

Kaip jau buvo minėta, pirmąjį procesą *init* sukuria kernelis, *init*'as kuria kitus procesus, o kiekvienas sukurtas procesas taip pat gali kurti naujus procesus. Gauname medį, kurio pradžioje *init* procesas. Procesas, kuriantis naują procesą vadinamas **tėvu** (*parent*), o sukurtas naujas procesas – **vaiku** (*child*).

Proceso būsenos (supaprastintai, skirtingose OS pilnas būsenų sąrašas skiriasi):

- *RUNNING* – šiuo momentu procesas vykdomas ant kurio nors CPU;
- *READY* – procesas laukia kol gaus CPU (kol OS jį pažadins);
- *WAITING* – procesas laukia kokio nors OS įvykio;
- *ZOMBIE* – pasibaigęs, bet iš procesų lentelės dar nepašalintas procesas (toks procesas jau nebedirbs, bet OS dar saugoma informacija apie jį).

## Procesų grupė

Procesų grupė (*process group*) – tai susijusių procesų rinkinys. Šis grupavimas skirtas palengvinti susijusių procesų valdymą (signalų siuntimą visiems grupės nariams). Procesų grupė identifikuojama numeriu – **PGID**. Kiekvienas procesas priklauso vienai procesų grupei. Naujai kuriamas procesas paveldi procesų grupę iš tėvo proceso. **Procesų grupės lyderis** (*PG leader*) – procesas, kurio PID

sutampa su PGID. Procesų grupė sunaikinama, kai paskutinis jos narys pasibaigia arba pereina į kitą procesų grupę.

PROCESŲ GRUPĖS PAVYZDYS: vykdant komandą `cat failas | wc` naujai sukurti procesai `cat` ir `wc` priklausys naujai procesų grupei, kurios lyderis bus `cat` procesas, o šią komandą vykdančią komandų interpretatorių liks senoje procesų grupėje.

## Sesija

Sesija (*session*) – procesų grupių rinkinys, į kurį įeina vienos prisijungimo sesijos procesų grupės. Sesija identifiikuojama numeriu – **SID**. Procesas priklauso tai sesijai, kuriai priklauso jo procesų grupė. Naujas procesas SID paveldi iš tėvo proceso. Procesas gali sukurti naują sesiją ir tapti šios **sesijos lyderiu** (*session leader*). Sesija sunaikinama, kai sunaikinama paskutinė į ją įeinanti procesų grupė.

### SESIJOS PAVYZDŽIAI:

- sesijai priklauso visi procesai paleisti iš vieno SSH prisijungimo; jei naudotojas prisijungia dar kartą (atidaro antrą SSH langą) – iš antro prisijungimo paleidžiami procesai priklausys kitai sesijai.
- vykdant komandą `cat failas | wc` nors ir bus sukurta nauja procesų grupė, tačiau visi procesai (komandų interpretatorių, `cat`, `wc`) liks toje pačioje sesijoje.

## Saugumo sistema

Taigi turime mažiausiai du saugumo lygius, kuriuose veikia OS programos:

1. *kernel mode* – privilegijuotas režimas, kuriuo veikia *kernel*'is;
2. *user mode* – apribotas režimas, kuriuo veikia visi OS procesai.

*init* procesas paleidžiamas naudotojo su *uid=0,gid=0* (*root*) teisėmis. Turėtų kilt klausimas, kaip atsiranda kitų naudotojų teisėmis veikiantys procesai? UNIX sistemoje tam naudojami *syscall*'ai *setuid()*, *seteuid()*, *setgid()*, *setegid()*, *setreuid()*, *setregid()* ir trys *uid/gid* identifikatorių poros:

1. *real* – naudotojo, paleidusio procesą UID;
2. *effective* (*euid/egid*) – naudotojo, kurio teisėmis šiuo metu dirba procesas UID;
3. *saved set-user-id* – išsaugota UID.

Tam tikslui naudojami *syscall*'ai, pvz.: norėdamas skaityti kažkokį failą, procesas vykdo specialią CPU instrukciją, perduodančią užklausą *kernel*'iui (įvyksta *software interrupt*).

Procesai taip pat gali keistis duomenimis tarpusavyje. Tam OS palaiko visą aibę *IPC* mechanizmų. Su kai kuriais iš jų jau susidūrėte, pvz. kanalas (*pipe*):

```
$ cat failas | wc
```

## Proceso atributai

Norint suvaldyti procesus, OS apie kiekvieną procesą turi būti saugoma informacija:

- būseną – RUNNING/READY/WAITING/...
- leidimai(?) (*credentials*) – realus ir efektyvus naudotojas ir grupė, užsaugotas *set-user-ID*, užsaugotas *set-group-ID*, papildomos grupės
- tapatybė – PID, PPID, PGID, SID, terminalas su kuriuo procesas surištas, komandos paleidusios procesą vardas, komandos argumentai

- informacija dispečeriui – prioritetas, dispečerio klasė, ...
- statistika – CPU naudojimo statistika, naudojamos vRAM dydis ir būseną, sunaudotas laikas (realus, user CPU, kernel CPU) ...
- ... – darbinis katalogas, šakninis katalogas, umask, naudojami failai, resursų limitai, virtualios atminties regionai, signalų lentelė, *syscall*'ų vektorius, *swap* informacija...

Ne visa saugoma informacija lengvai pasiekama naudotojui, be to skirtingose OS yra skirtumų (kitokios būsenos, kitoks dispečerio algoritmas, ir t.t.)

## Laiko apskaita

Su kiekvienu procesu susiję bent trys laikai skaičiuojami nuo proceso paleidimo iki dabartinio momento arba proceso pabaigos:

1. *realus (wall time)* – tikras/astronominis laikas nuo proceso paleidimo;
2. *user time* – kiek CPU laiko procesas gavo/sunaudojo savo skaičiavimams;
3. *system time* – kiek CPU laiko sugaišo *kernel*'is aptarnaudamas proceso *syscall*'us.

## Prioritetas

Dispečerio užduotis – optimaliai paskirstyti laiką procesams, tačiau kas yra „optimalu“ labai priklauso nuo konkretaus atvejo: aparatūros savybių, sprendžiamų uždavinių ir t.t (pvz.: skaičiavimo uždaviniai ir interaktyvios programos; laiko nuostoliai procesų perjungimui). Vienareikšmiškai geriausio dispečerio algoritmo dar nerasta. Todėl skirtingose OS naudojami dispečerio algoritmai skiriasi. POSIX'e standartizuoti du prioriteto valdymo būdai:

1. dispečerio politikos (*scheduler policy*) parinkimas  
( *sched\_setscheduler()*, *sched\_setparam()* *syscall*'ai);
2. *NICE* prioriteto reikšmė (naudojama dispečerio politikos pagal nutylėjimą algoritme).

Keisti proceso dispečerio politiką paprastai gali tik *root* naudotojas, ši politika labiau skirta realaus laiko procesų valdymui. Eiliniam naudotojui prieinamas tik *NICE* reikšmės keitimas. *NICE* reikšmė įtakoja procesui skiriamą laiko dalį, paprastai galimos reikšmės -20 .. 19 (kai kuriose OS iki 20). Kuo mažesnė *NICE* reikšmė, tuo didesnis proceso prioritetas. Paprastai eilinis naudotojas savo procesam galim priskirtą *NICE* reikšmę tikrai didinti (mažinant prioritetą). Naujas procesas paveldi tėvo proceso *NICE*.

## ps - procesų sąrašas

Komanda **ps** leidžia pamatyti informaciją apie operacinės sistemos vykdomų procesų būklę.

```
ps [-Aaefl] [-G grplist] [-g grplist] [-o format]... [-p pidlist] [-t ttylist] [-U userlist] [-u userlist]
```

Parinktys (ne visos):

- **-A** – išveda informaciją apie visus procesus;
- **-a** – išveda informaciją apie visus procesus, kurie susiję su terminalais ir nėra sesijų lyderiai;
- **-d** – išveda informaciją apie visus procesus, išskyrus sesijos lyderius;
- **-e** – tas pats, kaip **-A**;
- **-f** – rodo informaciją "pilnu" formatu;
- **-l** – rodo informaciją "ilgu" formatu;
- **-G grplist** – rodo informaciją apie procesus, kurių *realūs gid* nurodyti *grplist*;
- **-g grplist** – rodo informaciją apie procesus, priklausančius grupėms, nurodytoms *grplist* (*efektyvūs* GID); (PASTABA: šio argumento prasmė skiriasi įvairiose OS, žr. *man*);
- **-o format** – rodo informaciją *format* nurodytu formatu;
- **-p pidlist** – rodo informaciją tik apie *pidlist* nurodytus procesus (*pidlist*: kableliais arba tarpais atskirtas PID sąrašas);

- **-t ttylist** – rodo informaciją tik apie procesus, susijusius su nurodytais **ttylist** terminalais;
- **-U userlist** – rodyti informaciją tik apie procesus, kurių **realūs uid** nurodyti **userlist**;
- **-u userlist** – rodyti informaciją tik apie **userlist** nurodytų naudotojų procesus (**efektyvūs UID**).

formatas	antraštė	aprašymas
<b>s</b>	<b>S</b>	proceso būseną
<b>uid</b>	<b>UID</b>	efektyvus <i>uid</i>
<b>user</b>	<b>USER</b>	efektyvus vartotojo vardas
<b>ruid</b>	<b>RUID</b>	realus <i>uid</i>
<b>ruser</b>	<b>RUSER</b>	realus vartotojo vardas
<b>gid</b>	<b>GID</b>	efektyvus <i>gid</i>
<b>group</b>	<b>GROUP</b>	efektyvus grupės vardas
<b>rgid</b>	<b>RGID</b>	realus <i>gid</i>
<b>rgroup</b>	<b>RGROUP</b>	realus grupės vardas
<b>pid</b>	<b>PID</b>	proceso numeris
<b>ppid</b>	<b>PPID</b>	proceso tėvo numeris
<b>pgid</b>	<b>PGID</b>	procesų grupės, kuriai priklauso procesas, numeris
<b>sid</b>	<b>SID</b>	sesijos, kuriai priklauso procesas, numeris
<b>%cpu</b>	<b>%CPU</b>	paskutiniu metu proceso naudoto CPU laiko procentas
<b>sz</b>	<b>SZ</b>	proceso dydis virtualioje atmintyje 4K dydžio puslapiais
<b>vsz</b>	<b>VSZ</b>	proceso dydis virtualioje atmintyje kilobaitais
<b>ni</b>	<b>NI</b>	proceso prioritetą nusakanti reikšmė (didesnė reikšmė reiškia mažesnį prioritetą)
<b>pri</b>	<b>PRI</b>	proceso prioritetas (didesnė reikšmė reiškia mažesnį prioritetą)
<b>etime</b>	<b>ELAPSED</b>	nuo proceso paleidimo praėjęs laikas ([ [dd - ]hh : ]mm : ss, kur dd - dienų, hh - valandų, mm - minučių, ss - sekundžių skaičius)
<b>time</b>	<b>TIME</b>	bendras proceso sunaudotas CPU laikas

<b>start_time</b>	<b>START</b>	proceso paleidimo data arba laikas
<b>tty</b>	<b>TT</b>	su procesu susijusio terminalo vardas (įrenginio vardas arba numeris)
<b>comm</b>	<b>COMMAND</b>	komandos (failo) vardas, kurią paleidus buvo sukurtas procesas
<b>args</b>	<b>COMMAND</b>	komandos vardas ir jos argumentai
<b>wchan</b>	<b>WCHAN</b>	vardas <i>kernel</i> 'io funkcijos, kurioje "miega" procesas ("-", jei procesas vykdomas)

Linux OS stulpelio **S** prasmė:

- **R** – procesas yra šiuo metu vykdomas arba pasiruošęs vykdymui (*RUNNING/RUNNABLE*);
- **S** – įvykio laukiantis procesas (*SLEEPING*);
- **D** – nepertraukiamas laukimas, paprastai IO (*DISK SLEEPING*);
- **Z** – procesas-zombis (*ZOMBIE*);
- **T/t** – *STOP* signalu sustabdytas arba trasuojamas/debuginamas procesas (*STOPPED*).

## nice, renice - proceso prioriteto keitimas

Komandos **nice** ir **renice** atlieka analogiškus veiksmus, t.y. keičia proceso *NICE* atributą, tačiau **nice** skirta paleisti procesą su pakeista *NICE* reikšme, o **renice** – veikiančio proceso *NICE* atributo keitimui.

```
nice [-n increment] command [argument...]  
renice [-n] increment [-g|-p|-u] identifier...
```

Parinktys:

- **-n increment** – naujos *NICE* reikšmės pokytis ( t.y.  $newNICE = oldNICE + increment$  ), **increment** g.b. ir neigiamas (didinti prioritetą), tačiau didinti prioritetą gali tik privilegijuotas (pvz.: *root* naudotojo) procesas;
- **-g** – keisti nurodytos procesų grupės *NICE* (**identifier** interpretuojami, kaip procesų grupės numeriai);
- **-p** – keisti nurodytų procesų *NICE* (tas pats daroma pagal nutylėjimą, jei nenurodytos **-g** ir **-u** parinktys);
- **-u** – keisti visų nurodyto vartotojo procesų *NICE*. Šiuo atveju vartotojo **identifier** gali būti jo vardas (*login*) arba UID.

**command** nurodo paleidžiamą komandą, o **argument...** - jos argumentus. **identifier** nurodo keičiamo proceso, procesų grupės arba vartotojo ID numerius.

Pavyzdys:

```
$ ps -l  
  
F S    UID      PID     PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD  
0 S   1029   825651   825650  0  80   0  -  2937  -      pts/1    00:00:00 bash
```

```

0 R 1029 825656 825651 0 80 0 - 2703 - pts/1 00:00:00 ps

$ renice -n 5 -u 1029

1029 (user ID) old priority 0, new priority 5

$ ps -l

F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD
0 S 1029 825651 825650 0 85 5 - 2937 - pts/1 00:00:00 bash
0 R 1029 825669 825651 0 85 5 - 2703 - pts/1 00:00:00 ps

$ nice -n 7 bash

$ ps -l

F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD
0 S 1029 825651 825650 0 85 5 - 2937 - pts/1 00:00:00 bash
0 S 1029 825681 825651 0 92 12 - 2942 - pts/1 00:00:00 bash
0 R 1029 825684 825681 0 92 12 - 2704 - pts/1 00:00:00 ps

```

## top - interaktyvus procesų sąrašas

Nestandartinė komanda, skirtingose OS skiriasi. Įvairiose UNIX os yra ir daugiau panašias funkcijas atliekančių komandų (*prstat*, *htop*, *systat*, ...)

Komandos **top** parinktys:

- **-b** – paketinis režimas, t.y. **top** komanda periodiškai išvedinėja būseną (tinka pvz. kartu su **-n** norint **top** rezultatus įrašyti į failą);
- **-d secs** – Atnaujina **top** komandos išvedimą kas **secs** sekundžių;
- **-n number** – statistiką išveda **number** kartų ir pabaigia darbą;
- **-o fieldname** – rūšiuoti procesus pagal nurodytą lauką **fieldname**;
- **-u identifier** – rodomi tik naudotojo **identifier** (*login* arba *UID*) procesus;
- ... – daugiau **top** parinkčių galite rasti **man** puslapyje.

**top** komanda suteikia galimybę stebėti procesoriaus aktyvumą realiaame laike. Ji išveda informaciją apie tuos procesus, kurie intensyviausiai naudoja CPU, surūšiuodama šį sąrašą pagal CPU panaudojimą (arba kitą parinktyse nurodytą kriterijų). Išvedimas vyksta reguliariais laiko intervalais atsižvelgiant į tai, kas vyksta sistemoje. Pavyzdžiui:

Likusi išvedama informacija susijusi su individualiais procesais ir panaši į informaciją, išvedamą **ps** komandos: **PID** – proceso numeris, **USER** – proceso naudotojas, **PR** – proceso prioritetas, **NI** – prioritetą įtakojanti reikšmė, **VIRT** – bendras proceso dydis virtualioje atmintyje (kilobaitais), **RES** – proceso naudojamas fizinės atminties kiekis (kilobaitais), **S** – proceso būseną (žr. procesų būsenas **ps** komandos aprašyme, **I** – nieko nedarantis procesas), **%CPU** – kokią CPU laiko dalį procentais atitinkamas procesas naudoja paskutiniu metu, **%MEM** – proceso naudojama fizinės atminties dalis procentais, **TIME+** – bendras proceso sunaudotas CPU laikas, **COMMAND** – komandos/programos vardas, kurią vykdo procesas.

Paleista **top** komanda valdoma klaviatūra. Komandos:

- **q** – baigti darbą;

- **u** – rodyti tik vieno naudotojo procesus (**top** paprašo įvesti naudotojo vardą);
- **k** – „nužudyti“ procesą (paprašo įvesti PID);
- **r** – keisti *nice* reikšmę;
- ...

## strace - proceso išskviečiami *syscall'ai*

Nestandartinė komanda. Kitose UNIX OS tą patį daro *truss*, *ktrace/kdump*.

Komanda **strace** naudojama norint pamatyti programos išskviečiamus *syscall'us* ir siunčiamus bei gaunamus signalus.

```
strace { -c | [-frtttT] } [-e expr]... [-o file] [-p pid]... { -p pid | command [args] }
```

Parinktys (ne visos):

- **-c** – rodo tik iškviestų *syscall'u* statistiką;
- **-t, -tt, -ttt** – eilutės pradžioje rodo įvykio laiką nuo trasavimo pradžios;
- **-r** – eilutės pradžioje rodo įvykio laiką nuo prieš tai išvesto įvykio (ne laiką praleistą vykdant *syscall'ą*);
- **-T** – eilutės pradžioje rodo laiką, sugaištą apdorojant *syscall'ą*;
- **-p pid** – trasuoti **pid** nurodytus procesus;
- **-f** – trasuoti ir proceso sukurtus procesus (t.y. proceso "vaikus");
- **-e trace** – trasuoti tik **trace** nurodytus *syscall'us* (pagal nutylėjimą trasuojami visi, t.y. **-e all**);
- **-o filename** – trasavimo logą rašyti ne į terminalą, o į nurodytą failą **filename**.

Kiekviena **strace** išveda eilutę rodo proceso išskviečiamą *syscall'ą* arba signalą. Norint bendrai suprasti, ką daro procesas, nebūtina tiksliai žinoti kiekvienos eilutės prasmę. *Syscall'ai* smulkiau aprašomi savo **man** puslapiuose, pvz.: **open(2)**, **execve(2)** ir pan.

## kill - nutraukti procesą

**kill** komanda dažniausiai naudojama procesų nutraukimui.

```
kill [-s signal] pid...  
kill -l [signal]
```

Parinktys:

- **-s signal** – siunčiamo signalo numeris arba vardas;
- **-l** – parodyti signalų sąrašą (visų, arba tik su nurodytu numeriu/vardu).

**pid** nurodo, kam siunčiamas signalas:

- **>0** – procesui, kurio numeris nurodytas;
- **0** – visiems procesams, kurių procesų grupės numeris sutampa su **kill** proceso grupės numeriu;
- **-1** – visiems procesams (kuriems gali pasiūst **kill** procesas, pvz.: visiems komandą paleidusio naudotojo procesams);
- **<0** – signalas siunčiamas visiems tos procesų grupės procesams (procesų grupės numeris - absoliutinė PID reikšmė).

Jei signalas nenurodytas – siunčiamas *SIGTERM* signalas (jo numeris 15).

Pagal nutylėjimą, gavęs daugumą signalų procesas baigia darbą. Kai kuriuos signalus procesas gali ignoruoti (tame tarpe ir *SIGTERM*). Norint nutraukti į *SIGTERM* nereaguojantį procesą galima siųsti *SIGKILL* (9) signalą, kurio procesas negali ignoruoti. Procesas "užmigdomas" *SIGSTOP*, o žadinamas *SIGCONT* signalais.

## **^Z, fg, bg, jobs, kill** – komandos užmigdymas ir žadinimas

Šiame skyriuje **komanda** vadinama visa komandos eilutė, kuri gali būti sudaryta iš daugiau nei vienos programos iškviatimo (pvz.: **\$ cat file | wc**).

**^Z** klavišų kombinacija aprašyta [čia](#). Ši kombinacija siunčia vykdomos komandos procesams *SIGSTOP* signalą ir "atkabina" komandą nuo terminalo. Tokie procesai miegos, kol nebus nutraukti su **kill** arba pažadinti su **fg** arba **bg** komandomis. Komandų interpretatorius tokias sustabdytas užduotis atsimena ir kiekvienai jų priskiria numerį (*jobID*), komandose jis nurodomas **%N** (N - numeris).

**jobs** – parodo sustabdytų ar fone veikiančių užduočių sąrašą.

**fg [%N]** (*foreground*) komanda pažadina paskutinę užmigdytą arba **%N** nurodytą užduotį ir grąžina jai terminalą.

**bg [%N]** (*background*) komanda pažadina paskutinę užmigdytą arba **%N** nurodytą užduotį, bet negrąžina jai terminalo, t.y. užduotis veikia fone, jos išvedama informacija gali kartais būt matoma terminale, bet procesas bus sustabdytas (ne nutrauktas), jei bandys skaityti iš terminalo.

**kill %N** – nutraukti užduotį N.

Naudojimo pavyzdys ( **sleep N** komanda pamiega N sekundžių ir pasibaigia):

```
$ sleep 30
^Z
[1]+  Stopped                  sleep 30
$ bg
[1]+ sleep 30 &
$ sleep 20
^Z
[2]+  Stopped                  sleep 20
$ bg
[2]+ sleep 20 &
$ jobs
[1]-  Running                  sleep 30 &
[2]+  Running                  sleep 20 &
$ fg %1
sleep 30
```



```
[2] Done sleep 20
$
```

Čia paleidžiamos dvi **sleep** užduotys, abi atkabinamos nuo terminalo, parodomas užduočių sąrašas ir terminalas atiduodamas pirmajai.

## who – prie sistemos prisijungusių naudotojų sąrašas

Komanda **who** yra skirta peržiūrėti informacijai apie šiuo metu su sistema dirbančius vartotojus. Komanda išveda prisijungusių vartotojų vardus, spec. įrenginių vardus terminalams, per kuriuos prisijungę naudotojai, prisijungimo laiką ir kt.

```
who [-mTu] [-abdHlrt] [file]
who [-mu] -s [-bHlrt] [file]
who -q [file]
who am { i | I }
```

Iškvietimo formos su argumentais **am i** arba **am I** parodo tik informaciją apie terminalą iš kurio kviečiama komanda **who**.

Bendras komandos išvedimo formatas yra:

NAME	STATE	LINE	TIME	IDLE	PID	COMMENT	EXIT
------	-------	------	------	------	-----	---------	------

Išvedamos informacijos prasmė:

1. **NAME** – naudotojo prisijungimo vardas;
2. **STATE** – ar leidžiama kitiems naudotojams rašyti į terminalą (naudojant **write**, **talk** ar panašias komandas), galimos reikšmės: **+** - galima, **-** - negalima, **?** - neaišku;
3. **LINE** – terminalo spec. įrenginio failo vardas (kataloge **/dev**);
4. **TIME** – laikas, kada naudotojas prisijungė;
5. **IDLE** – kiek laiko terminalas neaktyvus/nenaudojamas, laikas nurodomas valandomis ir minutėmis, taškas (.) rodo, kad terminalas buvo naudojamas per paskutinę minutę;
6. **PID** – su terminalu susijusio shell proceso PID;
7. **COMMENT** – komentarų eilutė (turinys priklauso nuo OS)
8. **EXIT** – pasibaigusio, bet neperleisto iš naujo proceso pabaigos kodas

**PASTABA:** **state** perjungiama **msg** komanda.

Komandos **who** parinktys:

- **-a** – tas pat, kaip būtų nurodytos **-b**, **-d**, **-l**, **-p**, **-r**, **-t**, **-T**, **-u** parinktys;
- **-b** – paskutinios sistemos užkrovimo data (nuo kada sistema dirba);
- **-d** – parodo tik pasibaigusį, bet iš naujo nepaleistų procesų sąrašą (į šią informaciją galima nesigilinti, ji susijusi su **init**, **getty** ir **wait()** )
- **-H** – išveda stulpelių pavadinimus;
- **-l** – išveda tas terminalines linijas, kuriose sistema laukia kažkieno prisijungimo. Tokiu atveju vardo lauke matome žodį „LOGIN“, o kiti laukai yra tokie patys kaip ir išvedime apie prisijungusius vartotojus, išskyrus tai, kad nėra lauko **STATE**;
- **-m** – išveda informaciją tik apie einamąjį (iš kurio iškviesta komanda) terminalą;
- **-q** – rodo tik prisijungusių vartotojų vardus ir kiek yra prisijungusių vartotojų;
- **-r** – parodo **init** proceso veikimo lygį (**run-level**)
- **-s** – išvedami laukai: **name**, **line**, **time** (tas pats, kaip be parinkčių);

- **-T** – tas pats kaip ir **-s** parinktyje, bet papildomai dar išvedams laukas **STATE**;
- **-u** – rodo informaciją tik apie vartotojus, kurie tuo metu prisijungę.

## iostat – I/O statistika

*Nestandartinė komanda, skirtinguose UNIX skiriasi.*

Komanda **iostat** parodo I/O statistiką, t.y. *kernelio* turimą informaciją apie terminalo, blokinų įrenginių (diskų, juostų ir pan.) I/O apkrovą ir CPU panaudojimą.

```
iostat [ -c ] [ -d ] [ -k | -m ] [ -x ] [ interval [ count ] ]
```

Parinktys (ne visos):

- **-c** – rodo CPU panaudojimą įvairiuose režimuose procentais;
- **-d** – rodo diskų apkrovos statistiką;
- **-k** – diskų duomenų pralaidumas rodomas KB/sekundę;
- **-m** – diskų duomenų pralaidumas rodomas MB/sekundę;
- **-x** – rodo išplėstinę diskų statistiką.

Operandai:

- **interval** – sveikas skaičius, nurodantis kas kiek sekundžių išvesti statistiką.
- **count** – sveikas skaičius, nurodantis kiek kartų išvesti statistiką.

Komandai **iostat** nurodžius **interval**, pirma išvedama statistika apie visą laiką nuo OS paleidimo, o kiekvienas tolesnis išvedimas atspindi kitą, komandoje nurodytą laiko intervalą.

**iostat** komandos išvedamos informacijos laukų reikšmės pateiktos jos **man** puslapyje (**iostat(1)**).

## vmstat - virtualios atminties statistika

*Nestandartinė komanda, skirtinguose UNIX skiriasi.*

Komanda **vmstat** rodo OS virtualios atminties statistiką.

```
vmstat [-d | -s] [-S k|K|m|M] [delay [count]]
```

Parinktys (ne visos):

- **-d** – rodo diskų statistiką;
- **-s** – parodo kai kurių OS skaitiklių reikšmes (įvykių kiekį, laiką);
- **-S k|K|m|M** – pakeičia išvedamo vienetus į 1000 (k), 1024 (K), 1000000 (m) arba 1048576 (M) baitų.

**vmstat** komandos išvedamos informacijos laukų reikšmės pateiktos jos **man** puslapyje (**vmstat(8)**).

Naudojimo pavyzdžiai

```
$ vmstat
$ vmstat 1
$ vmstat -d 1 10
```

## mpstat - kiekvieno CPU statistika

*Nestandartinė komanda, skirtinguose UNIX skiriasi.*

Komanda **mpstat** išveda kiekvieno CPU statistiką atskirai.

```
mpstat [ interval [ count ] ]
```

**mpstat** komandos išvedamos informacijos laukų reikšmės pateiktos jos **man** puslapyje (**mpstat(1)**).

Naudojimo pavyzdžiai

---

```
$ mpstat  
  
$ mpstat 1  
  
$ mpstat 1 10
```

## sar - sistemos apkrovimo statistika

*Nestandartinė komanda, skirtinguose UNIX skiriasi.*

Ši komanda irgi renka sistemos statistiką, tačiau, skirtingai nei ankstesnės, gali veikti dvejais režimais:

- interaktyviu, kai nurodytu intervalu išvedama OS informaciją;
- skaityti duomenis iš anksčiau surinktos į failą statistikos (šį failą gali formuoti **sysstat** paketo servisas arba interaktyviu režimu paliesta **sar** komanda).

```
sar [-BbdqrSuvWy] [ -f [ filename ] | -o [ filename ] [ -i interval ] [ -s [ hh:mm[:ss] ] ]  
] [ -e [ hh:mm[:ss] ] ] [ interval [ count ] ]
```

Parinktys (ne visos):

- **-f filename** – skaityti duomenis iš anksčiau suformuoto duomenų (dvejetainio) failo;
- **-o filename** – išsaugoti duomenis į dvejetainį failą;
- **-s time** – dominančios statistikos pradžios laikas (kai skaitoma iš failo);
- **-e time** – dominančios statistikos pabaigos laikas (kai skaitoma iš arba rašoma į failą);
- **-i interval** – duomenis išveda kaip galima artimesniais intervalais nurodytam **interval** sekundžių skaičiui;
- **-B, -b, -d, -q, -r, -S, -u, -v, -W, -y** – išveda įvairias skaitiklių grupes (*syscall*'ai, diskai, swap'inimas, CPU, eilių ilgiai ir t.t.), jau matytas aukščiau aprašytose komandose.

Operandai:

**interval** – kokiais intervalais spausdinti statistiką (sveikas skaičius, sekundės);  
**count** – kiek kartų kartoti spausdinimą (jei nenurodyta - spausdinama vieną kartą).

Naudojimo pavyzdžiai

---

Interaktyvus

```
$ sar 1 10  
  
$ sar -d 1 10
```

```
$ sar -o sar.bin 1 30
```

Skaitymas iš failo:

```
$ sar -f sar.bin  
$ sar -f sar.bin -i 5  
$ sar -f sar.bin -i 5 -d  
$ sar -s 10:00 -e 14:00
```

## ps

- pagal komandos **ps -efl** rezultatus išsiaiškinkite:
  - kokia komanda paleistas procesas turintis PID=1?  
  
init
  - koks/kokie procesai/komandos yra paruošti vykdymui arba jau vykdomi (Linux abu šiuos atvejus žymi viena busena)?  
  
Running
  - Koks jūsų paleisto proceso PID, PPID? Koks procesas yra jūsų paleisto proceso "tėvu" ir "seneliu"? Koks bendras **ps** proceso virtualios atminties dydis?

Vaikas yra tas kurio ppid sutampa su kito proceso pid

- Koks iš matomų procesų yra sunaudojęs daugiausiai CPU laiko?
- Kaip su **ps** išvesti visų procesų sąrašą, kuriame matytųsi tik: proceso savininkai, PID, "tėvo PID", procesų grupė, sesija, terminalas, komandos vardas?

Ps -o

- Atsidarykite du terminalo langus. Vienaime paleiskite **man ls**. Kitame su **ps** komanda išsiaiškinkite savo procesų hierarchiją:
  - kiek ir kokias komandas vykdančių procesų turite;
  - kuris procesas kurį paleido (nusipaišykite procesų medį);
  - kokias procesų grupes ir sesijas turite ir kurie procesai yra jų lyderiai.
- suraskite visus SID sistemoje;
- suraskite didžiausios ne nulinės sesijos (turinčios daugiausia procesų) procesus;
- suraskite didžiausios procesų grupės procesus;
- suraskite seniausią veikiančią procesus.

## top

- pabandykite **top**;
- vieną kartą išveskite į ekraną 10 procesų, kurie naudoja daugiausiai CPU laiko (tai galima atlikti interaktyvia komanda, kurios aprašymą rasite **top(1) man** puslapyje);
- top** , shift+f

- shift+m memory
- shift+p cpu
- shift+n procesas id
- shift+t time
- kuris(-ie) procesas(-ai) paruošti vykdymui arba jau vykdomi CPU?
- paleiskite **top**, kad procesai būtų surūšiuoti pagal **VIRT** stulpelį;
- paleiskite **top** komandą, kad ji rodytų tik jums priklausančius procesus. Kiek %CPU laiko maksimaliai naudoja jūsų paleisti procesai?
- nenutraukdami **top** komandos atidarykite dar vieną terminalo langą ir jame paleiskite vykdymui programą **cksum /dev/zero**. Kiek %CPU laiko naudoja jūsų paleista **cksum**?
- užmigdykite (nesvarbu koku būdu) **cksum** ir patikrinkite, kiek %CPU ji naudoja;
- atgaivinkite **cksum** ir patikrinkite, kiek %CPU ji naudoja;
- nutraukite **cksum**.
- pabandykite paleisti keletą CPU apkraunančių procesų:
  - paleiskite keturis procesus, vykdančius komandą **cksum /dev/zero** (galite atidaryti atskirus terminalus arba pasinaudoti komandų vykdymu *background*);
  - paleiskite **top**, kuri rodytų tik jūsų procesus;
  - su **top** vieno **cksum** NICE reikšmę nustatykite į **19**;
  - kaip NICE reikšmė paveikė procesą?
- *[advanced]* kodėl dviejų **cksum** procesų **193.219.36.233** serveryje gali neužtekti, kad pamatytumėte NICE poveikį?

## strace

- išbandykite **strace** su **/usr/bin/id** komanda, pabandykite surasti *syscall*'ą, kuris spausdina **id** rezultatus į terminalą;
- kelis kartus pratraukite **/bin/true**:
  - gaunate vienodus ar skirtingus rezultatus? Jei skiriasi - kas?
  - ar *syscall*'ai arba jų iškviatimo tvarka kinta?
  - kokius failus naudoja (ar bando naudoti) **/bin/true**?
- paleiskite **nano** ir **strace** taip, kad dirbdami su **nano** (rinkdami tekstą, skaitydami ir išsaugodami failus) lygiagrečiai matytumėte šio proceso kviečiamus *syscall*'us.

## Informacija apie prisijungusius vartotojus

- kas šiuo metu prisijungė prie sistemos (kokie loginai)?
- kada perkrauta sistema?
- pabandykite "pašnipinėti" ka veikia kuris nors vienas naudotojas (jo procesai, procesų medis, vykdomos komandos, ką tos komandos daro ...).

## SUID/SGID

- kataloge **/data/ld/ld1** turime du vykdomuosius failus: **id1** ir **id2**:
  - palyginkite šiuos failus su **/usr/bin/id** (ir turinį, leidimus, ...). Kuo jie skiriasi?
  - paleiskite kiekvieną failą. Kodėl skiriasi išvedama informacija?
- pabandykite atsikopijuoti **id** į savo katalogą ir nustatyti SUID/SGID:
  - pabandykite paleisti originalą ir kopiją. Ar skiriasi originalo ir kopijos išvedama informacija?
  - duokite kolegai (iš jo userio) paleisti jūsų kopiją;
  - pabandykite paleisti kolegos **id** kopiją su SUID/SGID. Ar skiriasi originalo ir kopijos išvedama informacija?
  - *[Advanced]* kodėl pats nematote SUID/SGID efekto?

## Sistemos statistikos užduotys

- įvertinkite sistemos apkrovimą:

- sukasi tuščiai, arti 100% apkrovos (programos sukasi pilnu greičiu, bet viena kitai netrukdo) ar programos konkuruoja dėl kažkurio resurso (diskai, CPU ...) ir veikia lėčiau, nei galėtų;
  - iš ko apie tai sprendžiat?
  - kaip manote, kaip turėtų atrodyti kitos dvi sistemos būsenos (ir kaip jas atpažinti)?
- išsiaiškinkite, kurią dieną per paskutines 7 dienas sistema buvo labiausiai apkrauta pirmoje dienos pusėje (nuo 8:00 iki 12:00). Duomenų failai kaupiami [/var/log/sa](#).

**Sar -f /var/log/sa -s 8:00 -e 12:00**