

[Advanced] Sugalvokite ir išbandykite komandą, kuri išvestų `/sbin` katalogo turinį puslapiais.

`ls /sbin | more`

Sugalvokite ir išbandykite komandą, kuri `/sbin` katalogo turinį išvestų ne į ekraną, bet į failą `sbin.txt`.

`ls /sbin > sbin.txt`

Iškvieskite paskutinių 5 vykdytų komandų sąrašą.

`history 5`

Iš vykdytų komandų sąrašo iškvieskite paskutinę vykdytą komandą, kurioje būtų panaudota simbolių seka "qbd".

`!?qbd`

Kaip sužinoti, kuriame vadovėlio skyriuje aprašyta C funkcija "mkdir"? .

`man -k mkdir`

Kaip atversti C funkcijos "mkdir" detalesnį aprašymą (t.y. šią funkciją aprašantį man puslapį)?

`man -s2 mkdir`

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]** Peržiūrėdami failą `sbin.txt` parinkite tokias parinktis, kad failą galėtume peržiūrėti:

- po vieną įrašą;
 - `more -1 sbin.txt`
- po tris įrašus.
 - `more -3 sbin.txt`

[Advanced] Kaip peržiūrėti pirmas dvi failų `sbin.txt` ir `/data/ld/ld1/tekstas1` eilutes?

`head -2 sbin.txt /data/ld/ld1/tekstas1`

Kaip peržiūrėti paskutines dvi failo `sbin.txt` eilutes? (sugalvokite 2 komandos užrašymo variantus)

`tail -2 sbin.txt`

`tail -n -2 sbin.txt`

Kaip peržiūrėti failo `sbin.txt` eilutes nuo 2-osios iki pabaigos?

`tail +2 sbin.txt`

Kaip išvesti paskutinius 15 failo `sbin.txt` baitų? (sugalvokite 2 komandos užrašymo variantus)

`tail -c -15 sbin.txt`

`tail -15c sbin.txt`

[Advanced] pabandykite pasekti, kaip keičiasi/auga `/var/log/messages` failas (į failą rašomi įvairūs sisteminiai pranešimai, tarp jų ir informacija apie naujus prisijungimus prie serverio per SSH)

`tail -3f /var/log/messages`

- Failo pradžioje įterpkite eilutę "Šiame faile yra info apie `/sbin` katalogo turinį".
- Nukopijuokite šią eilutę į failo pabaigą.

`alt + 6` nukopijuoja eilutę

Nustatykite, kurioje eilutėje, stulpelyje ir kelintas yra įrašo "arp" simbolis "p"

kursorius ant simbolio ir tada `ctrl + c`. ats: line 3, col 3

[Advanced] Atidarykite failus `sbin10.txt` bei `sbin10mod.txt` (kad jie redaktoriuje būtų atidaryti vienu metu) bei pabandykite pereiti nuo vieno failo prie kito. Uždarykite abu failus

`nano sbin10.txt sbin10mod.txt PEREITI I KITA FAILA SU ALT+rodykle`

- **[Advanced]** Atlikite veiksmų seką:
 - į failą `f1.txt` įrašykite tekstą "tekstas" ir užsaugokite failą
 - įvykdyskite:
 - `$ ln f1.txt f2.txt`
 - kieta nuoroda iš `f2.txt` į `f1.txt`
 - `$ ln f1.txt f3.txt`

- kieta nuoroda is f3.txt i f1.txt
- \$ rm f1.txt
-
- į f2.txt įrašykite tekstą "bilekas" ir užsaugokite failą
- kuris tekstas matomas faile f3.txt ir kodėl?
- nes jie yra susieti kieta nuoroda. Kol bus bent viena nuoroda, failas bus pasiekiamas

Išsiaiškinkite kas yra /usr/lib, /dev/null, /bin/true, /etc/passwd failai.

file /usr/lib -> /usr/lib: directory

file /dev/null -> /dev/null: character special (1/3)

file /etc/passwd -> /etc/passwd: Unicode text, UTF-8 text

kuo skiriasi /lib/libdl-2.33.so /lib/libdl.so.2 /lib/ld-2.33.so (ar tai tokio pat tipo failai, ar ne).

file libdl-2.33.so === libdl-2.33.so: ELF 32-bit LSB shared object, Intel 80386, version 1 (SYSV), dynamically linked, for GNU/Linux 3.2.0, stripped

file libdl.so.2 === libdl.so.2: symbolic link to libdl-2.33.so

file ld-2.33.so ===== ld-2.33.so: ELF 32-bit LSB shared object, Intel 80386, version 1 (GNU/Linux), static-pie linked, stripped

išbandykite strings su parinktimi -d ir be jos failui /bin/true. Kodėl skiriasi rezultatai?

Skiriasi dėl to, kad naudojant komandą su -d bus analizuojama tik vykdomojo failo dalis, kuri bus įkelta

išbandykite strings failui /etc/netconfig (tekstinis failas). Ar išvedamos visos jo eilutės? Kaip priversti strings išvesti visas eilutes?

strings -1 /etc/netconfig

[Advanced] su od komanda išsiaiškinkite simbolio kodą, kuris naudojamas tekstiname /etc/hosts faile eilutės pabaigai žymėti.

od -tCc /etc/hosts ===== 0a

[Advanced] kaip koduojama š raidė faile [source:basics|iconv/lt-u8.txt](#) ?

c5

[Advanced] kuo skiriasi [source:basics|iconv/lt-u8.txt](#) ir [source:basics|iconv/lt-u8w.txt](#) failai?

lt-u8w.txt failas turi 3 nematomus simbolius pradžioje

Parašykite grep filtrus, kurie atrinktų eilutes:

turinčias simbolį 'c'

grep c tekstas.txt

turinčias simbolių seką 'aa'

grep aa tekstas.txt

neturinčias simbolių 'b'

grep -v b tekstas.txt

neturinčias nei didžiųjų, nei mažųjų 'a'

grep -iv a tekstas.txt

Failo /etc/passwd:

suraskite savo prisijungimo vardą aprašančią eilutę;

grep lukkuz1 /etc/passwd

suraskite visas savo grupės studentų prisijungimo vardus aprašančias eilutes;

grep IFF-1/6 /etc/passwd

suraskite savo bendravardžių prisijungimo vardus aprašančias eilutes.

grep luk /etc/passwd

kiek eilučių faile /etc/passwd?

wc -l /etc/passwd

kiek failų turite savo namų kataloge?

ls | wc -w

12

kiek failų yra /bin/ kataloge?

```
ls /bin | wc -w
```

```
111
```

surūšiuokite /etc/passwd.

```
sort /etc/passwd
```

surūšiuokite /etc/passwd pagal trečią lauką (laukai šiame faile skiriami :).

```
sort -t : -k 3,3 /etc/passwd
```

[Advanced] surūšiuokite atvirkščia tvarka /bin katalogo turinį pagal kietų nuorodų skaičių. (Reikia žinoti ls -l formatą)

```
ls -l /bin | sort -t " " -k 2,2
```

sudarykite IP adresų sąrašą be iš eilės einančių pasikartojimų

```
cut -f 1 -d " " /data/ld/ld1/Solaris_access_log | uniq
```

sudarykite IP adresų sąrašą be pasikartojimų (kiekvienas IP sąrašė gali pasikartoti tik vieną kartą)

```
cut -f 1 -d " " /data/ld/ld1/Solaris_access_log | uniq -c -u
```

[Advanced] su cmp palyginkite failus /data/ld/ld1/ls ir /data/ld/ld1/ps. Jie vienodi ar skiriasi?

vienodi, nes nieko neįveda

[advanced] remiantis tuo ką radote, kaip manote, kurios dar failus reikėtų atidžiai patikrinti?

Isilauzėlis pakeitė command kintamąjį, kuris sugadintam faile rodo I laikina isilauzėlio failą. Tada tas failas yra nukopijuojamas į sbin katalogą

Sudalykite jį į keturis failus maždaug po 250KB, kurių vardai prasidėtų simboliais vv..

```
split -b 250k /data/ld/ld1/file_1M vv
```

Pasitikrinkite gautą rezultatą (patikrinkite failų dydžius).

```
ls -l -h rodo gerus dydžius
```

Sujunkite gautus keturis failus į vieną failą vvv.

```
cat vvaa vvab vvac vvad vvae >> vvv
```

Patikrinkite, ar gautas vvv ir pradinis file_1M vienodi.

```
diff /data/ld/ld1/file_1M vvv
```

patikrinkite, ar /bin kataloge yra vienodų failų, t.y. pavadinimas gali skirtis, bet turinys vienodas (patikrinkite bent 15 paskutinių abėcėlės tvarka failų).

```
cksum $(ls /bin | sort | tail -15) | uniq -w 12 -d
```

[Advanced] Duoti 3 failai su vienodu lietuvišku tekstu, bet juose naudojamos skirtingos koduotės:

1. UTF-8 [source:basics | iconv/lt-u8.txt](#)
2. ISO-8859-4 [source:basics | iconv/lt-i4.txt](#)
3. ISO-8859-13 [source:basics | iconv/lt-i13.txt](#)

• [Advanced] išsiaiškinkite kurį galite matyti normaliai (t.y. kurio koduotė atitinka jūsų terminalo koduotę)

```
lukkuz1@oslinux ~ $ cat lt-u8.txt.1
```

Lietuviškas tekstas:

ąčėėjšųž

ĄČĘĖJŠŲŽ

```
lukkuz1@oslinux ~ $ cat lt-i4.txt
```

Lietuviškas tekstas:

???

Suraskite didžiausią `/etc` subkatalogą.

~~Sukurkite failą savo namų kataloge ir į jį įrašykite vieną simbolį. Kiek vietos jis užima?~~

~~Nukopijuokite aukščiau sukurtą failą į `/tmp/` katalogą. Kiek vietos užima ši failo kopija?~~

~~[advanced] jei namų ir `/tmp` kataloguose failai užima skirtingai vietos, kodėl taip yra?~~

~~[advanced] sukurkite failą `testas` naudodami komandą `dd if=/dev/zero of=testas bs=1k seek=1024 count=1`. Koks jo dydis? Kiek vietos jis užima? Pabandykite `testas` failą nusikopijuoti. Koks dabar dydis ir užimama vieta? Patikrinkite, ar originalo ir kopijos turinys toks pat. Kaip manote, kame reikalas? Hint: Skylėtieji failai ([Sparse files](#))~~

Kokioje failų sistemoje yra:

jūsų namų katalogas

`df`

katalogai `/var` ir `/tmp`

`df /var df /tmp`

katalogas `/run`

`df /run`

Kiek vietos išnaudojote ir kiek dar turite laisvos savo namų kataloge?

`df`

```
Filesystem    1K-blocks    Used Available Use% Mounted on
/dev/root      51289568 8999092 39652704 19% /
```

- pagal komandos `ps -efl` rezultatus išsiaiškinkite:

- kokia komanda paleistas procesas turintis `PID=1`?

`ps -efl`

```
F S UID      PID  PPID C PRI NI ADDR SZ WCHAN  STIME
```

```
TTY      TIME CMD
```

```
4 S root      1    0 0 80  0 - 586 -    2022 ?    00:00:49 init [3]
```

- koks/kokie procesai/komandos yra paruošti vykdymui arba jau vykdomi (Linux abu šiuos atvejus žymi viena busena)?

Tie procesai kurie pažymėti `R` yra paruošti vykdymui arba jau vykdomi.

- Koks jūsų paleisto proceso `PID`, `PPID`? Koks procesas yra jūsų paleisto proceso "tėvu" ir "seneliu"? Koks bendras `ps` proceso virtualios atminties dydis?

```
PID: 3456052, PPID: 3455234 (tevo( 0 S lukkuz1 3455234 3455233 0 80 0
- 2999 -    16:32 pts/32 00:00:00 -bash senelio) . 2727
```

- Koks iš matomų procesų yra sunaudojęs daugiausiai CPU laiko?

```
1 R root      7    2 0 80  0 -  0 -    2022 ?    00:59:03 [rcu_sched]
```

- Kaip su `ps` išvesti visų procesų sąrašą, kuriame matytųsi tik: proceso savininkai, `PID`, "tėvo `PID`", procesų grupė, sesija, terminalas, komandos vardas?

- `lukkuz1@oslinux ~ $ ps -o uid,pid,ppid,gid,sid,tt,comm`

- `UID PID PPID GID SID TT COMMAND`

- `2151 3455234 3455233 100 3455234 pts/32 bash`

- `2151 3457422 3455234 100 3455234 pts/32 ps`

- `dauluk@oslinux ~ $`

- Atsidarykite du terminalo langus. Viename paleiskite `man 1s`. Kitame su `ps` komanda išsiaiškinkite savo procesų hierarchiją:

- kiek ir kokias komandas vykdančių procesų turite;

`ps -U lukkuz1`

- kuris procesas kurį paleido (nusipaišykite procesų medį);

```

3455220
3455233
3455234
3457761
3457477
3457479
3457480
3457483
3457495

```

- o kokias procesų grupes ir sesijas turite ir kurie procesai yra jų lyderiai.

- suraskite visus SID sistemoje;

```
lukkuz1@oslinux ~ $ ps -o pgid,sid -A
```

```
PGID  SID
```

```
1    1
0    0
0    0
0    0
```

.....

- suraskite didžiausios ne nulinės sesijos (turinčios daugiausia procesų) procesus;

```
ps -efl --sort=-sid
```

- suraskite didžiausios procesų grupės procesus;

```
ps -efl --sort=-gid
```

- suraskite seniausią veikiantį procesus.

```
Lukkuz1 ~ $ ps -efl --sort=-time
```

```

F S UID      PID  PPID C PRI NI ADDR SZ WCHAN  STIME TTY      TIME CMD
0 R paudom2 3424164    1 77 80 0 - 2487 -   13:39 ?    02:48:49 cksum /dev/zero
0 R paudom2 3424200    1 77 80 0 - 2487 -   13:39 ?    02:48:34 cksum /dev/zero
0 R paudom2 3424301    1 77 80 0 - 2488 -   13:40 ?    02:48:13 cksum /dev/zero
0 R paudom2 3424299    1 77 80 0 - 2488 -   13:40 ?    02:48:04 cksum /dev/zero
4 S root    1545     1 0 80 0 - 49890 ?    2022 ?    01:12:40 /usr/bin/vmtool
1 S root      7      2 0 80 0 - 0 -    2022 ?    00:59:03 [rcu_sched]
0 R andmin  3451856    1 65 80 0 - 2488 -   16:11 ?    00:42:28 cksum /dev/zero
0 R andmin  3451891    1 64 80 0 - 2488 -   16:11 ?    00:42:12 cksum /dev/zero
0 R andmin  3451874    1 64 80 0 - 2488 -   16:11 ?    00:42:08 cksum /dev/zero
0 S fraali  3040096    1 0 92 12 - 2924 -   Feb15 ?    00:18:58 top -b
1 S zabbix   2206    2168 0 80 0 - 2123 -   2022 ?    00:16:26 /usr/sbin/zabbi
0 S matras  3128529    1 0 92 12 - 2918 -   Feb17 ?    00:13:34 top -b

```

- 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);

```
lukkuz1 ~ $ top -b -n 1 -o %CPU | head -17
```

- kuris(-ie) procesas(-ai) paruošti vykdymui arba jau vykdomi CPU?

Pazymėti R

- paleiskite **top**, kad procesai būtų surūšiuoti pagal VIRT stulpelį;

```
lukkuz1 $ top -b -n 1 -o VIRT | head -17
```

- paleiskite **top** komandą, kad ji rodytų tik jums priklausančius procesus. Kiek %CPU laiko maksimaliai naudoja jūsų paleisti procesai?

```
Lukkuz1 ~ $ top -b -n 1 -o TIME+ -u lukkuz1
```

- 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?
 - Nice ir renice
- *[advanced]* kodėl dviejų `cksum` procesų 193.219.36.233 serveryje gali neužtekti, kad pamatytumėte NICE poveikį?

1. Kokie šios komandos visi galimi naudojimo variantai?

```
/usr/bin/wc [-c | -m | -C ]
```

`wc -c`, `wc -m`, `wc -C`

2. Kokios komandos pagalba galima suskaičiuoti žodžių kiekį faile paskutinėms 10-čiai failo eilučių?

```
Lukkuz1 ~ $ tail -10 sbin.txt | wc -w
```

10

3. Duotas failas `data.txt`. Suskaičiuokite kiek šiame faile yra unikalių eilučių (atsakymui pateikite naudotą komandą ir gautą skaičių)

```
wc -l data.txt
```

4. Išveskite sistemos proceso, turinčio daugiausia vaikų-procesų kiekį, id ir jo vaikų-procesų skaičių (atsakymui pateikite komandą)

```
ps -eo pid,ppid | awk '{print $1}' | sort | uniq -c | sort -rn | head -1
```