

Protsessid

https://wiki.itcollege.ee/index.php/Linux/Unix_protsessid

[https://en.wikipedia.org/wiki/Process_\(computing\)](https://en.wikipedia.org/wiki/Process_(computing))

https://wiki.itcollege.ee/index.php/Cron_ja_crontab

Iga programmikood, mis jookseb Linuxis süsteemis, on protsess. Linuxis protsesse võib jagada kolmeks: interaktiivsed protsessid, automaatsed protsessid ja teenused. Interaktiivsed protsessid on need, mida käivitab kasutaja ja mis on seotud terminaliga (omavad mingit vastastikmõju kasutajaga). Neid saab omakorda jagada kaheks: taustaprotsessid ja esiplaaniprotsessid.

Taustaprotsessid (*background processes*) on need, mis jooksevad tagataustal ja antud hetkel kasutajaga otseselt ei suhtle. Need võivad olla kahes olekus: ajutiselt peatatud (*paused*) ja jooksvad (*running*). Taustaprotsessi võib vaadata käsu **jobs** abil.

Esiplaaniprotsessid (*foreground processes*) on protsessid, millega kasutaja just praegusel hetkel suhtleb ning mis kasutavad oma standardväljundiks ja -sisendiks terminali.

Automaatsed protsessid on need, mis käivituvad automaatselt. Nad ei ole seotud terminaliga, pigem on nad pandud FIFO (*first-in, first-out*) tüüpi järjekorda. Kasutatakse ajastatud toiminguid: Linuxis CRON, MS Windowsis Scheduled Tasks.

Teenused e. süsteemiprotsessid (daemons) on niisugused protsessid, mis jooksevad pidevalt tagataustal, mis ei suhtle kasutajaga terminali kaudu ning pakuvad kasutajale mingit teenust. Tavaliselt käivitatakse teenused arvuti alglaadimisel. Nendeks on näiteks sshd, apache jne.

Protsesside atribuudid

Igal protsessil on kindel hulk atribuute, mis teda kirjeldavad. Nendeks on:

1. Protsessi ID e. PID (Process Identifier): unikaalne identifitseeriv number, mis viitab ühele protsessile.
2. Vanema ID e. PPID (Parent Process Identifier): protsessi PID, mis käivitas antud protsessi.
3. Nice number: protsessi „sõbralikkuse“ määr teiste protsesside vastu. Seda numbrit kasutatakse protsessi prioriteedi arvutamiseks protsessori poolt.
4. Terminal e. TTY: terminal, millega protsess seotud on.
5. Reaalse ja efektiivse kasutaja kasutajanimi (RUID ja EUID): protsessi omanik. RUID on kasutaja, mis käivitas antud käsku, EUID on kasutaja, mis määrab süsteemiressursside õigused. Tavaliselt langevad nad kokku.

Protsesside olekud

https://en.wikipedia.org/wiki/Process_state

Running - protsess on süsteemis hetkel töötav või see on valmis hakkama tööle st. protsess ootab enda määramist mõnele süsteemi CPU'le.

Waiting - protsess ootab mõnda teatud sündmust või ressursi.

Stopped - protsess on peatatud tavaliselt vastu võetud signaaliga. Veasilumis(debug) režiimis olev protsess võib olla ka peatatud(stopped) seisundis.

zombie – vanemaga seose kaotanud mittetöötav protsess, võib olla võimalus tappa vanemprotsess (PPID – Parent Process ID) ja nii ka zombie-protsess sulgeda

protsesside haldamiseks: top, htop, ps, atop → paigaldada htop kui veel ei ole

Protsesside jälgimine

Protsesside jälgimiseks kasutame peamiselt rakendusi ps ja top. Neid käskke kasutatakse protsesside visualiseerimiseks. ps omab erinevaid atribuute, mida kombineerides, saab erinevate võimalustega

kuvada protsesse. Kui seda käivitada ilma parameetriteta, siis kuvab see ainult käesoleva shelli infot. Selle väljund näeb välja umbes selline:

```
PID TTY
TIME CMD
10361 pts/0 00:00:00 su
10362 pts/0 00:00:00 bash
11152 pts/0 00:00:00 ps
```

Et näha kõiki käigus olevaid protsesse koos kasutatud ressursidega, tuleb kasutada käsku `ps aux`. Selle väljund näeb välja järgmine:

```
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 1 0.0 0.1 2036 700 ? S 11:59 0:01 init [2]
root 2 0.0 0.0 0 0 ? SN 11:59 0:00 [ksoftirqd/0]
root 3 0.0 0.0 0 0 ? S 11:59 0:00 [watchdog/0]
```

Kõikide protsesside vaatamine:

```
ps -e
ps -ef
ps -eF
ps -ely #näitab ära ka PPID
```

Lisainfo (vt lõpus näited):

```
man ps
```

Antud haldusvahend väljastab vaid ühekordselt praegusel hetkel ressurside infot. Et pidevalt jälgida protsesse, kasutatakse rakendust `top`.

top (reaalajas)

<https://wiki.itcollege.ee/index.php/Top>

h abiinfo

```
Z,B,E,e Global: 'Z' colors; 'B' bold; 'E'/'e' summary/task memory scale
l,t,m Toggle Summary: 'l' load avg; 't' task/cpu stats; 'm' memory info
0,1,2,3,I Toggle: '0' zeros; '1/2/3' cpus or numa node views; 'I' Irix mode
f,F,X Fields: 'f'/'F' add/remove/order/sort; 'X' increase fixed-width
```

```
L,&,<,> . Locate: 'L'/'&' find/again; Move sort column: '<'/'>' left/right
R,H,V,J . Toggle: 'R' Sort; 'H' Threads; 'V' Forest view; 'J' Num justify
c,i,S,j . Toggle: 'c' Cmd name/line; 'i' Idle; 'S' Time; 'j' Str justify
x,y . Toggle highlights: 'x' sort field; 'y' running tasks
z,b . Toggle: 'z' color/mono; 'b' bold/reverse (only if 'x' or 'y')
u,U,o,0 . Filter by: 'u'/'U' effective/any user; 'o'/'0' other criteria
n,#,^0 . Set: 'n'/'#' max tasks displayed; Show: Ctrl+'0' other filter(s)
C,... . Toggle scroll coordinates msg for: up,down,left,right,home,end
```

```
k,r Manipulate tasks: 'k' kill; 'r' renice
d or s Set update interval
W,Y Write configuration file 'W'; Inspect other output 'Y'
q Quit
```

(commands shown with '.' require a visible task display window)

Press 'h' or '?' for help with Windows,

Type 'q' or <Esc> to continue

- . Use multiple windows, each with separate config opts (color,fields,sort,etc)
 - . The 'current' window controls the Summary Area and responds to your Commands
 - . that window's task display can be turned Off & On, growing/shrinking others
 - . with NO task display, some commands will be disabled ('i','R','n','c', etc)
 - until a different window has been activated, making it the 'current' window
 - . You change the 'current' window by: 1) cycling forward/backward; 2) choosing a specific field group; or 3) exiting the color mapping or fields screens
 - . Commands available anytime -----
 - A . Alternate display mode toggle, show Single / Multiple windows
 - g . Choose another field group and make it 'current', or change now by selecting a number from: 1 =Def; 2 =Job; 3 =Mem; or 4 =Usr
 - . Commands requiring 'A' mode -----
 - G . Change the Name of the 'current' window/field group
 - * a , w . Cycle through all four windows: 'a' Forward; 'w' Backward
 - * - , _ . Show/Hide: '-' Current window; '_' all Visible/Invisible
- The screen will be divided evenly between task displays. But you can make some larger or smaller, using 'n' and 'i' commands. Then later you could:
- * = , + . Rebalance tasks: '=' Current window; '+' Every window
(this also forces the current or every window to become visible)

In 'A' mode, '*' keys are your essential commands. Please try the 'a' and 'w' commands plus the 'g' sub-commands NOW. Press <Enter> to make 'Current'

vt ka man top

sorteerimise muutmine:

f #avab väljade seadistamise

nooleklahviga valida soovitud veerg

s #määrab valitud (aktiivse) veeru uueks sorteerimise aluseks

q #väljub sätetest

R #sorteerimise järjestuse muutmine

mitut protsessi näidata?

n #võimaldab määrata kas n arv või kõik (0 on piiramatult)

veergude muutmine:

f, nooleklahvidega valida välja, nool paremale liigutab, nool vasakule lülitab liigutamise välja, Esc lõpetab. Lisada näiteks PPID enne PID'i ja salvestada see muudatus (vaikimisi läheb ~/.toprc), et ka järgmisel top'i käivitamisel oleks see kohe olemas. Tühik või d lülitab veeru kuvamist.

pikad read

nooleklahvidega vasakule, paremale saab vaadata pikki ridu, saab kasutada kursori liigutamise klahve (nooleklahvid, PageUp, PageDown) kogu protsessitabeli vaatamiseks

W #salvestab sätted ~/.toprc faili

R #lülitab sorteerimist kasvav/kahanev

u → student + Enter #kuvab kasutaja student protsessid

u + Enter #kuvab kõikide kasutajate protsessid

L #otsimine. Otsida: ssh (kui on paigaldatud OpenSSH server ja protsess peaks töötama). Kui tööjaam siis otsida: bash

k #tapab protsessi, võimalik ka signaali väärtus öelda, signaalid vt kill -l

käivitada nano ja leida protsess üles ning sulgeda see protsess

d #värskendamise aeg (delay)

V #puuvaade (Forest view)

c #lülitab vaadet Cmd name/line - võimaldab lülitada töötavate protsesside käskude ja täispikkade otsiteekondade vahel

C #lülitab kerimise koordinaate

i #lülitab jõude olevaid protsesse - võimalus vaadata vaid hetkel aktiivseid protsesse ja tabeli pikkust vähendada

S #lülitab protsessi töötamise kestvuse kuvamist

l #lülitab koormust päises

t #lülitab protsess/CPU päises

m #lülitab mäluinfot päises

J #veergudes olevate numbrite joonduse lülitamine

j #käsu (command) joondus veeru ulatuses

0 #lülitab nulle

z #värvilise režiimi lülitamine

A #lülitab mitme akna režiimi, sh näha ka PPID

a, w #lülitab kordamööda erinevaid aknaid: vt ülaserivas tähist

n #võimaldab määrata, mitu protsessi näidatakse

i #muudab aktiivse valiku suurust

z #muudab aktiivse valiku värvi, võimalus kas üks või mitu osa värviliseks muuta

htop (reaalajas)

<https://wiki.itcollege.ee/index.php/Htop>

https://viki.pingviin.org/Hallates_CPU_protsesse_htop_abiga

täiustatud versioon top'ist

vaikimisi ei pruugi olla paigaldatud (sudo apt update && sudo apt install htop && sudo apt clean)

värvilisem, enamus Midnight Commander'i klahvikombinatsioonid

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
921	root	20	0	21860	1536	1184	R	1.0	0.3	0:00.30	htop
1	root	20	0	24032	2100	1316	S	0.0	0.4	0:01.25	/sbin/init
281	root	20	0	17096	640	448	S	0.0	0.1	0:00.32	upstart-udev-bridg
284	root	20	0	21580	1452	776	S	0.0	0.3	0:00.24	udev --daemon
373	syslog	20	0	52860	1476	1088	S	0.0	0.3	0:00.14	rsyslogd -c5
375	syslog	20	0	52860	1476	1088	S	0.0	0.3	0:00.00	rsyslogd -c5
376	syslog	20	0	52860	1476	1088	S	0.0	0.3	0:00.02	rsyslogd -c5
370	syslog	20	0	52860	1476	1088	S	0.0	0.3	0:00.21	rsyslogd -c5
379	messageb	20	0	24152	952	684	S	0.0	0.2	0:00.20	dbus-daemon --syst
489	root	20	0	7124	600	120	S	0.0	0.1	0:00.00	dhclient3 -e IF_ME
531	root	20	0	21576	1012	340	S	0.0	0.2	0:00.00	udev --daemon
533	root	20	0	15048	380	192	S	0.0	0.1	0:00.04	upstart-socket-bri
578	root	20	0	21576	752	80	S	0.0	0.1	0:00.00	udev --daemon
611	root	20	0	4180	636	536	S	0.0	0.1	0:00.01	/sbin/getty -8 384
614	root	20	0	4180	636	536	S	0.0	0.1	0:00.02	/sbin/getty -8 384
619	root	20	0	4180	632	536	S	0.0	0.1	0:00.01	/sbin/getty -8 384
621	root	20	0	4180	636	536	S	0.0	0.1	0:00.00	/sbin/getty -8 384
623	root	20	0	4180	636	536	S	0.0	0.1	0:00.00	/sbin/getty -8 384
628	daemon	20	0	16776	384	224	S	0.0	0.1	0:00.00	atd
630	root	20	0	18976	888	696	S	0.0	0.2	0:00.01	cron
713	root	20	0	54316	1604	1184	S	0.0	0.3	0:00.10	/bin/login --
762	student	20	0	28260	7372	1724	S	0.0	1.5	0:01.44	-bash
860	root	20	0	37044	1780	1312	S	0.0	0.4	0:00.11	sudo -i

ps (process status)

<https://wiki.itcollege.ee/index.php/Ps>

<https://viki.pingviin.org/Ps>

kuvab Enter'ile vajutamise hetkeseisuga protsessid

ps aux #BSD stiil
ps waux #w – wide output
ps -ef #SystemV [init](#)'iga UNIX'id

grep'i abil saab filtreerida

sorteerimine top'is:

ava top

F

liikuda nooleklahvidega üles/alla soovitud valiku peale

s #määrab uue sorteerimise kriteeriumi

ESC või q läheb tagasi top'i

ps -ef | grep root

ps aux | grep root

näited vt man ps → EXAMPLES

- **ps** käsk ilma lisavõtmeteta näitab ainult põhilist informatsiooni sinu kasutajanime all jooksvatest protsessidest:

```
root@server:~# ps
  PID TTY          TIME CMD
  713 tty1        00:00:00 login
  860 tty1        00:00:00 sudo
  861 tty1        00:00:00 bash
  916 tty1        00:00:00 ps
root@server:~# _
```

PID tulp näitab protsessi ID numbrit, TTY tulp näitab protsessi terminalikuuluvust, TIME tulp näitab protsessoriaega mida protsess on kasutanud ning CMD tulp näitab käsu nimetust mis parajasti töötab.

- **ps -f** käsk kuvab täisformaadis protsessitabeli sinu arvutis jooksvate protsesside kohta:

```
root@server:~# ps -f
UID      PID  PPID  C  STIME TTY          TIME CMD
root      713    1    0  16:41 tty1        00:00:00 /bin/login --
root      860   762    0  16:42 tty1        00:00:00 sudo -i
root      861   860    0  16:43 tty1        00:00:00 -bash
root      917   861    0  16:44 tty1        00:00:00 ps -f
root@server:~# _
```

UID tulp näitab kasutaja ID numbrit, PPID näitab vanemprotsessi ID'd, STIME näitab kellaaega millal protsess käivitati.

- **ps -ef** käsuga kuvatakse kõikide kasutajate protsessid täisformaadis:

```

root@server:~# ps -ef
UID      PID  PPID  C  STIME TTY          TIME CMD
root      1    0    0  16:41 ?        00:00:01 /sbin/init
root      2    0    0  16:41 ?        00:00:00 [kthreadd]
root      3    2    0  16:41 ?        00:00:00 [ksoftirqd/0]
root      4    2    0  16:41 ?        00:00:00 [kworker/0:0]
root      5    2    0  16:41 ?        00:00:00 [kworker/u:0]
root      6    2    0  16:41 ?        00:00:00 [migration/0]
root      7    2    0  16:41 ?        00:00:00 [cpuset]
root      8    2    0  16:41 ?        00:00:00 [khelper]
root      9    2    0  16:41 ?        00:00:00 [netns]
root     10    2    0  16:41 ?        00:00:00 [sync_supers]
root     11    2    0  16:41 ?        00:00:00 [bdi-default]
root     12    2    0  16:41 ?        00:00:00 [kintegrityd]
root     13    2    0  16:41 ?        00:00:00 [kblockd]
root     14    2    0  16:41 ?        00:00:00 [ata_sff]
root     15    2    0  16:41 ?        00:00:00 [khubd]
root     16    2    0  16:41 ?        00:00:00 [md]
root     17    2    0  16:41 ?        00:00:00 [khungtaskd]
root     18    2    0  16:41 ?        00:00:00 [kworker/u:1]
root     19    2    0  16:41 ?        00:00:00 [kswapd0]
root     20    2    0  16:41 ?        00:00:00 [ksmd]
root     21    2    0  16:41 ?        00:00:00 [fsnotify_mark]
root     22    2    0  16:41 ?        00:00:00 [ecryptfs-kthrea]
root     23    2    0  16:41 ?        00:00:00 [crypto]
root     31    2    0  16:41 ?        00:00:00 [kthrotld]
root     32    2    0  16:41 ?        00:00:00 [kworker/0:1]
root     33    2    0  16:41 ?        00:00:00 [scsi_eh_0]

```

Protsesside haldamine

Kuidas luua protsesse, on selge – tuleb lihtsalt sisestada käsk, mis käivitab mingit programmi. Selleks aga, et protsessi katkestada, tuleb kasutada käsk `kill`, `killall` vajalike parameetritega. Antud käsk ei tee midagi muud, kui saadab protsessile mingi signaali – signaali valib kasutaja ise. Käsk `kill` saadab spetsiifilise signaali spetsiifilisele protsessile või protsessigrupile. Käsk `kill` tapab protsesse protsessi PID (Process ID) järgi kasutades lisaks signaalivõtit mis spetsifitseerib kuidas protsess tapetakse.

- Antud käsk saadab spetsiifilisele protsessile SIGTERM signaali, mis lubab protsessil teha viimased hädavajalikud toimingud või salvestada andmed, kuigi võib juhtuda et antud signaal ei ole piisav protsessi sulgemiseks:

```
kill -15 protsessi_ID
```

- Antud käsk saadab spetsiifilisele protsessile SIGKILL signaali, mis ei luba teha protsessil viimase hetke toiminguid ning sulgeb selle viivitamatult:

```
kill -9 protsessi_ID
```

Signaalid

https://wiki.itcollege.ee/index.php/Signaalid_ja_kill

<https://wiki.itcollege.ee/index.php/Killall>

```
kill -l #vaatame signaale
```

```
signaalide numברי teisendamiseks: kill -l <number>, N: kill -l 15
```

SIGTERM 15 Termineerib protsessi „viisakalt“. Ei garanteeri, et protsessi töö lõpetatakse.

SIGINT 2 Katkestab protsessi. Protsess võib seda signaali ignoreerida.

SIGKILL 9 Katkestab „jõuga“ protsessi. Protsess ei saa tööd jätkata.

SIGHUP 1 Peatab protsessi ja käivitab uuesti. Kasutatakse peamiselt

teenuste taaskäivitamiseks, kui nad töötavad valesti või kui konfiguratsioonifail muutus.

vaikimisi saadetakse signaal 15 (SIGTERM), mis on viisakas protsessi sulgemine (ootab kuni protsess lõpetab)

```
kill <PID> #protsessi viisakas sulgemine signaaliga SIGTERM
kill -9 <PID> #SIGKILL signaal (9)
kill -s KILL <PID> #nimeliselt signaali SIGKILL (9) saatmine protsessile
kill -KILL <PID> #nimeliselt signaali SIGKILL (9) saatmine protsessile
```

```
pidof <protsessinimi> #leiab töötava protsessinumbri nime järgi
kill $(pidof <protsessinimi>) #kombineerimine
```

```
killall <protsessinimi> #protsessi sulgemine signaaliga SIGTERM nime järgi
killall -9 <protsessinimi> #SIGKILL signaal (9)
killall -s KILL <protsessinimi> #nimeliselt signaali SIGKILL (9) saatmine protsessile
killall -KILL <protsessinimi> #nimeliselt signaali SIGKILL (9) saatmine protsessile
```

signaalinimi võib ka väikeste tähtedega olla kirjutatud

CTRL+C saadab SIGINT (2) signaali, et sulgeda protsess
CTRL+Z saadab SIGTSTP (20) signaali (paus), taastamine SIGCONT (18) signaaliga, mis saadetakse käsuga fg (*resume in foreground*). SIGTSTP on sarnane SIGSTOP signaaliga selle erinevusega, et seda ei saa ignoreerida või hallata.

lisainfo <http://man7.org/linux/man-pages/man7/signal.7.html>
man 7 signal (vt man -f signal)

nice – protsessi prioriteet (-20 kõrgeim; 19 madalaim, vaikimisi 0), reguleerib protsessoriaega (kõrgema prioriteediga saab rohkem), lisalugemist <https://wiki.itcollege.ee/index.php/Nice> programmis top:

```
* r :Renice-a-Task
    You will be prompted for a PID and then the value to nice it to.
```

```
Entering no PID or a negative number will be interpreted as the default
shown in the prompt (the first task displayed). A PID value of zero means
the top program itself.
```

```
A positive nice value will cause a process to lose priority. Conversely,
a negative nice value will cause a process to be viewed more favorably by
the kernel. As a general rule, ordinary users can only increase the nice
value and are prevented from lowering it.
```

```
If you wish to abort the renice process, do one of the following depending
on your progress:
```

- 1) at the pid prompt, type an invalid number
- 2) at the nice prompt, type <Enter> with no input
- 3) at any prompt, type <Esc>

init - <https://wiki.itcollege.ee/index.php/Init>

Harjutus

signaalide vaatamine: kill -l

signaalide numbri teisendamine tähenduseks: kill -l <number>, N: kill -l 15

- Käivitage programm top taustal.

- top&
- Suunake töötavate protsesside nimekiri faili ps.txt
 - ps > ps.txt
- Suunake sama faili lõppu protsesside nimekiri koos detailse protsesside nimekirjaga
 - ps ef >> ps.txt
- Suunake töötavate tööde nimekiri faili jobs.txt
 - jobs -r > jobs.txt
- Saatke konkreetsele programmile term signaal protsessi numbri alusel
 - kill 14155
- Käivitage tekstiredaktor nano ja suruge see taustale
 - on seotud konkreetse terminalisessiooniga
 - nano
 - teises terminalis vaatame kas töötab: ps -e | grep nano
 - CTRL+Z (peatame töö ja saadame taustale)
 - jobs (vaatame kas on taustal & peatatud)
 - tagasitoomiseks fg 1 (kui on esimene jobs nimekirjas), sõltuvalt distrost:
 - fg %1
- Sulgege töö (nano) signaaliga TERM
 - killall -s TERM nano
- Käivitage korraldus cowsay (vajadusel paigaldada) või /usr/games/cowsay (kui games pole PATH muutujas). Cowsay sisend võtke failist /etc/issue ja väljund suunake faili /var/tmp/say.txt
 - cat /etc/issue | LC_ALL=C cowsay > /var/tmp/say.txt
- Käivitage korraldus cowsay ja võtke sisend failist /var/tmp/say.txt ning suunake väljund programmi less sisendisse
 - cat /var/tmp/say.txt | LC_ALL=C cowsay | less
- filtreerida ainult etteantud kasutaja protsessid, nt student
 - top -> u, student, enter
 - ps -u student

Küsimused

- Mis vahe on KILL ja TERM signaalil?
 - KILL sulgeb jõuga, TERM viisakalt (laseb lõpetada pooleliolevad tegevused, sulgeb protsessiga seotud andmed, tegevused)
- Miks ei ole mõistlik mõnda teenust, näiteks andmebaasiteenus sulgeda KILL signaali abil?
 - tegevused jäävad lõpetamata
- Mis juhtub protsessiga, mis saab signaali, mida ta ei töötle (signaali handler programmi pole)?
 - üldjuhul ignoreeritakse (vt man 2 signal)
- Selgitage, mida hoitakse keskkonnamuutujas PATH?
 - otsiteekonnad

* * *

graafiliselt: xkill (soovitav käivitada terminalist, katkestamiseks CTRL+C); System Monitor

Linuxis saab keskkonnamuutujate vaatamiseks kasutada korraldust env või ka printenv

<https://wiki.itcollege.ee/index.php/Env>

Lisavõimalused:

`set / less #kestprogrammi - ja keskkonnamuutujad, kohalikud muutujad, kestprogrammi funktsioonid`

`set -o posix; set / less #kestprogrammi - ja keskkonnamuutujad`

MS Windows'is

CTRL+SHIFT+ESC või ka Super+R->taskmgr

Process Explorer annab parema ülevaate:

<https://docs.microsoft.com/en-us/sysinternals/downloads/process-explorer>

Keskkonnamuutujad

https://wiki.itcollege.ee/index.php/Keskkonna_muutujad #erinevad OSid

<https://help.ubuntu.com/community/EnvironmentVariables>

otsiteekonna muutuja PATH täiendamine:

`~/ .profile`

`PATH="$HOME/bin:$HOME/.local/bin:$PATH"`

ehk siis:

`export PATH=$PATH:/kataloog/uuskaust`

`source ~/ .profile`

Harjutus: loome keskkonnamuutuja

luua kasutajale student uus muutuja nimega osadmin, mis koosneb kahest olemasolevast: masinanimi kasutajanimi

masinanime saame kuvada:

`hostname`

`echo $(hostname)`

kasutaja:

`echo $USER`

kokku:

`export osadmin="$(hostname) $USER"`

või kasutame tühiku paojada (*escape sequence*)

`export osadmin=$(hostname)\ $USER`

rakendamiseks käivitame sessiooniskripti:

`source ~/ .profile`

vajadusel muutuja tühistamiseks (mitu võimalust):

`unset osadmin`

`export osadmin=`

vt https://en.wikipedia.org/wiki/Escape_sequence

<http://www.keeleveeb.ee/dict/speciality/aks/dict.cgi?word=escape+sequence&lang=en>

kuvada muutuja ka standardväljundisse (*stdout*) ehk terminali ja suunata selle väärtus kasutaja kodukataloogi faili osadmin.txt

Kas seda loodud muutujat saab ka root kasutada? Põhjendada?

Kuidas korrektselt logida ühest kasutajast teiseks? Miks see on oluline?

Muutuja muutmine, loomine (üks neist)

globaalselt: /etc/profile, /etc/environment, /etc/bash.bashrc

kasutajapõhiselt: ~/.profile (tavaliselt olemas, soovitav), ~/.bashrc, ~/.bash_profile (kui ~/.bash_profile ei eksisteeri siis loetakse ~/.bash_login sisse); kõikidele uutele kasutajatele panna vastav fail /etc/skel/ kausta (nt: /etc/skel/.profile)

paigaldada rakendus cowsay ja teha nii, et see käivituks nii kasutaja root kui student alt ilma otsiteekonda ette andmata

Harjutus

kasutajana student:

* laadida alla skript shellshare (vt <https://enos.itcollege.ee/~edmund/materials/shellshare/>)

* paigutada kausta ~/.programmid/shellshare/

* teha skript kausta ~/.programmid/ ja määrata käivitavaks

* teha nii, et kaust ~/.programmid/ asuks kasutaja student otsiteekonnas (path) ehk siis saaks otse käivitada

* edastada oma terminali eetrise shellshare.net või siis linux.itcollege.ee kaudu

Mis vahe on kui muutujat tehakse alljärgnevalt? Selgita.

muutuja=väärtus

export muutuja=väärtus

tagada, et kasutaja student lokaat oleks eesti (et_EE.UTF-8), teha kindlaks, mis lokaadid süsteemis on. Märksõnad: locale, dpkg-reconfigure, localectl ...

vt ka https://wiki.archlinux.org/index.php/Keyboard_configuration_in_console

/usr/share/locale/

Logida teise kasutajana sisse (student → root) ja uurida keskkonnamuutujaid ning leida keskkonnamuutujate kaudu info (uid ja kasutajanimi), kes oli eelmine kasutaja kellest kasutajaks root ning kuvada see ühel real

Tööd

Tegemist on taustalolevate protsessidega.

<https://wiki.itcollege.ee/index.php/Jobs>

Harjutus

luua fail vorgud.txt (nt nano abil)

alustada lause kirjutamisega: võrguliidesed on

panna nano taustale

vaadata (veenduda), et nano töötab taustal ja rohkem midagi ei ole taustal käimas

vaadata süsteemist järgi, millised võrguliidesed on

jätta meelde või suunata need tekstifaili

tulla tagasi nano'sse failimuutmise juurde

kirjutada lause lõpuni

salvestada, väljuda, kuvada faili sisu stdout'i

Võrguliikluse jälgimine

iftop

iptraf-ng

kasutatud käsud

which #kus asub käsku käivitav binaarfail või skript
file #info failitüübi kohta
less #pikkade failide vaatamine
killall -l (näitab signaalide loetelu)
man 7 signal (ülevaade signaalidest), vt man -f signal

Viited

https://bash.cyberciti.biz/guide/Sending_signal_to_Processes
<http://www.howtogeek.com/107217/how-to-manage-processes-from-the-linux-terminal-10-commands-you-need-to-know/>
<http://serverfault.com/questions/41959/how-to-send-jobs-to-background-without-stopping-them>
<http://stackoverflow.com/questions/625409/how-do-i-put-an-already-running-process-under-nohup>
<https://wiki.itcollege.ee/index.php/Cowsay>
https://wiki.itcollege.ee/index.php/J%C3%B5udluse_j%C3%A4lgimine_ja_probleemilahendus_k%C3%A4surea_utiliitide_abil
<http://serverfault.com/questions/41959/how-to-send-jobs-to-background-without-stopping-them>
<https://linuxjourney.com/lesson/job-control>
<https://linuxjourney.com/lesson/stderr-standard-error-redirect>
<http://www.computerhope.com/unix/ujobs.htm> - jobs
<http://unix.stackexchange.com/questions/87745/what-does-ic-all-c-do>
<http://stackoverflow.com/questions/2499794/how-can-i-fix-a-locale-warning-from-perl>
<http://askubuntu.com/questions/162391/how-do-i-fix-my-locale-issue>
<http://unix.stackexchange.com/questions/181433/can-signal-be-ignored-lost>
<https://major.io/2010/03/18/sigterm-vs-sigkill/>

mahukate väljundite edastamine <https://help.ubuntu.com/community/Pastebinit>