Failiõigused

loengu slaidid

Siin dokumendis juttu õigustest UNIX'ilaadsetes OS'ides (sh Linux'is, macOS'is jne)
Vahepeal ka grep'i kasutamisest, mis abiks õiguste kui ka vajalike failide, kaustade kuvamisel.

http://askubuntu.com/questions/152001/how-can-i-get-octal-file-permissions-from-command-line

```
stat -c '%A %a %n' *
```

%A õigused inimloetavalt
%a õigused kaheksandsüsteemis
%n faili, kausta nimi

vt `man stat`

http://kuutorvaja.eenet.ee/kasutamine/os/failioigused.html
http://enos.itcollege.ee/~edmund/osadmin/materials/failioigused-UNIXis/failioigused-UNIXis.html

https://wiki.itcollege.ee/index.php/Linux/Unix_failiõigused
https://wiki.itcollege.ee/index.php/Setuid_/_Setgid eriõigused
https://wiki.itcollege.ee/index.php/Chmod õiguste muutmine
https://wiki.itcollege.ee/index.php/Chown omaniku muutmine
https://wiki.itcollege.ee/index.php/Umask vaikimisi loodavate failide, kataloogide õiguste muutmine

Õiguste määramine tähtedega
r(ead) - lugeda
w(rite) - kirjutada
e(x)ecute - käivitada

isiku määramine tähtedega
u(ser) - faili omanik
g(roup) - faili gruppi kuuluv kasutaja
o(ther) - mingi muu kasutaja süsteemis, kuulub faili omanikust erinevasse gruppi

õigused ja isikud koos:

```
    u
    g + r
chmod   - w  fail1 fail2 ...
    o = x
    a
```

kus u - user; g - group, o - others, a - all, r - read, w – write, x -execute

Plussmärk lisab, miinus eemaldab, võrdusmärk määrab konkreetse õiguse. Ühe märgi järele võib mitu õigust korraga kirjutada:
`u+rwx` #kasutajale lisatakse lugemine, kirjutamine, käivitamine

Samuti võib mitu isikut korraga kirjutada:
`ug+rwx` #kasutajale ja grupile lisatakse lugemine, kirjutamine, käivitamine

Loome harjutamiseks tühja faili hetke asukohta:
`>fail.txt`

Faili, kataloogi olemasolu kontrollimiseks ja õiguste vaatamiseks saame kasutada käsku ls:
ls #kuvab kausta sisu kompaktse loeteluna
ls -l #kuvab kausta sisu detailse loeteluna (-l)
ls -la #kuvab kausta sisu detailse loeteluna koos peidetud failide näitamisega (-a)

Lisame omanikule lugemine, kirjutamine, käivitamine (7); grupile ja teistele lugemine, käivitamine (5):
`chmod u+rwx,go+rx fail.txt`
… sulgudes on õigus kaheksandarvuna
Õiguste lisamisel olemasolevaid õigusi ei muudeta, uued lisatakse olemasolevatele.

Määrame (kirjutame vanad õigused üle) omanikule lugemine, kirjutamine, käivitamine (7); grupile ja teistele lugemine, käivitamine (5):
`chmod u=rwx,go=rx fail.txt`

Failidel ei peaks üldiselt käivitusõigust olema (turvarisk), eemaldame selle kõikidele:
`chmod a-x fail.txt`
.. seda võib ka kehtestada, vanu õigusi üle kirjutades:
`chmod u=rw,go=r fail.txt` #kaheksandarvuna 644 – sellised õigused peaks olema ka failidel veebiserveris

Kui isikut ei määra siis rakendatakse õigused kõikidele (omanik, grupp, ülejäänud):
`chmod -x fail.txt` #kõikidelt eemaldab käivitusõiguse
`chmod =r fail.txt` #määrab kõikidele vaid lugemisõiguse
`chmod +r fail.txt` #olemasolevatele õigustele lisatakse kõikidele vaid lugemisõigus

Sarnaselt kataloogiga:
`mkdir proov` #loome harjutamiseks kausta
`chmod =x proov/` #kui puudub lugemisõigus siis ei saa seal kataloogis anda käsku ls
`chmod u=rwx,go=rx proov/` #omanik saab kõike teha, grupp ja ülejäänud saavad kuvada sisu ja kausta siseneda ehk siis kaheksandarvuna 755 – sellised õigused peaks olema ka kataloogidel veebiserveris

Õigused tähtedega ja numbriliselt (saab vaadata ka mitme faili, kausta kohta korraga):

```
  stat -c '%A %a %n' fail.txt proov/
drwxr-xr-x 755 proov/
-rw-r--r-- 644 fail.txt
```

%A õigused inimloetavalt
%a õigused kaheksandsüsteemis
%n faili, kausta nimi
vt `man stat`

NB! Tähtedega õiguste lisamine ei muuda olemasolevaid õigusi kui lisame või eemaldame! See on väga oluline kui meil on vaja õigusi lisada ja olemasolevaid mitte muuta.
`chmod g+w fail.txt` #lisame grupile kirjutusõiguse antud failile
`chmod g-w fail.txt` #eemaldame grupilt kirjutusõiguse antud failile
`chmod g=w fail.txt` #määrame grupile ainult kirjutusõiguse antud failile (siin kirjutatakse olemasolevad õigused uutega üle)
`chmod o+w fail.txt` #lisame kõikidele teistele kirjutusõiguse antud failile
`chmod o-w fail.txt` #eemaldame kõikidelt teistelt kirjutusõiguse antud failile
`chmod o=w fail.txt` #määrame kõikidele teistele kirjutusõiguse antud failile (siin kirjutatakse olemasolevad õigused uutega üle)

kaustad:
`chmod g+rx proov/` #lisame grupile vaatamis- ja sisenemisõiguse antud kaustale
`chmod g-rx proov/` #eemaldame grupilt vaatamis- ja sisenemisõiguse antud kaustale
`chmod g=rx proov/` #määrame grupile ainult vaatamis- ja sisenemisõiguse antud kaustale (siin kirjutatakse olemasolevad õigused uutega üle)
`chmod o+rx proov/` #lisame kõikidele teistele vaatamis- ja sisenemisõiguse antud kaustale
`chmod o-rx proov/` #eemaldame kõikidelt teistelt vaatamis- ja sisenemisõiguse antud kaustale
`chmod o=rx proov/` #määrame kõikidele teistele vaatamis- ja sisenemisõiguse antud kaustale (siin kirjutatakse olemasolevad õigused uutega üle)

Õigused tähtedega ja numbriliselt (saab vaadata ka mitme faili, kausta kohta korraga):

```
  stat -c '%A %a %n' fail.txt proov/
drwxr-xr-x 755 proov/
-rw-r--r-- 644 fail.txt
```

%A õigused inimloetavalt
%a õigused kaheksandsüsteemis
%n faili, kausta nimi
vt `man stat`

grep’i kasutamisest

Kausta sisu kuvamise käske saame kombineerida omakorda näiteks grep’iga, mis aitab vähendada infomüra ja kiiremini leida otsitavat (filtreerida):
`ls | grep otsitav` #kus “otsitav” asemele kirjutada otsitava faili, kataloogi nimi või nimeosa, analoogselt saab ka teiste käskude väljundit filtreerida grep’i abil. See sõna on tõstutundlik (*case sensitive*) ehk siis otsitakse täpselt sellise tähesuurusega sümbolite jada.

Käsku grep võib kasutada ka korduvalt:
`ls | grep o1 | grep o2` #kus “o1” on esimene otsitav sümbolite jada ja “o2” teine

kuvame vaid terved (eraldi esinevad) sõnad:
`ls | grep -w o1`

kuvame tõstutundetult (nii suur- kui väiketähed)
`ls | grep -i o1` #kuvatakse nii “O1” kui “o1”

mitut parameetrit saab ka koos kuvada:
`ls | grep -iw o1` #kuvatakse nii “O1” kui “o1”, mis esinevad vaid eraldi sõnadena

kuvame read, mis ei sisalda otsitavat:
`ls | grep -v o1` #kuvatakse read, mis ei sisalda “o1”

lisaks voo filtreerimisele grep suudab otsida ka faili seest:
`grep “otsitav sümbolite jada” fail.txt`
… faili asukoha võib anda ka absoluutse aadressina (alates juur- või kodukataloogist)
näiteks serveris otsime tööjaama sisselogimisi IP-aadressi järgi:
`grep 192.168.0.2 /var/log/auth.log`

kõik teatud tüüpi failid:
`grep “otsitav sümbolite jada” *.txt` #otsitakse kõikidest .txt tüüpi failidest

lisainfo: `man grep`

Õiguste määramine numbritega

Kuna failisüsteemis vastab igale üksikule õigusele üks bitt, siis kõneldakse vahel failiõiguste asemel faili *loabittidest*.

```
chmod 755 fail.txt
```
… kus 755 on õigusi kirjeldav kaheksandarv

0 - SetUID
$7 = 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 4 + 2 + 1 = 7$
$5 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 4 + 0 + 1 = 5$
$5 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 4 + 0 + 1 = 5$

0 eriõigused (hetkel puuduvad ja sellisel juhul ei kirjutata tavaliselt)
4 lugemisõigus
2 kirjutamisõigus
1 käivitamisõigus

esimene arv (7) määrab õigused omanikule
… teine arv (5) grupile
… kolmas arv (5) ülejäänutele kes ei ole ei omanik ega grupp (ei kuulu gruppi)

Failide puhul üldiselt käivitusõigust ei anta – see tähendab turvariski kuna failis võivad sisalduda kahjulikud käsud, mida faili käivitadades saab rakendada. UNIX'ilaadsetes OSides saab selliselt mistahes failile käivitusõigust anda. Tavaliselt antakse käivitusõigus skriptile (erilise esimese reaga (shebang https://wiki.itcollege.ee/index.php/Shebang) käske sisaldav tekstifail) või binaarfailile (masinkood, tekib tavaliselt kompileerimise käigus kus kõrgkeeles kirjutatud programm tõlgitakse masinkoodi). Kõik arvutifailid võib jagada kahte kategooriasse: binaarfailid ja tekstifailid

Õigused tähtedega ja numbriliselt (saab vaadata ka mitme faili, kausta kohta korraga):
```
  stat -c '%A %a %n' fail.txt proov/
drwxr-xr-x 755 proov/
-rw-r--r-- 644 fail.txt
```

%A õigused inimloetavalt
%a õigused kaheksandsüsteemis
%n faili, kausta nimi
vt `man stat`

õigused lähevad tagant poolt ettepoole

failid
failiõiguste vaatamiseks:
```
ll fail.txt
ls -l fail.txt
stat -c '%A %a %n' fail.txt
```

```
chmod 6 fail.txt #sisuliselt 006 või o=rw
```
-------rw- 6 fail.txt

```
chmod 66 fail.txt #sisuliselt 066 või go=rw
```

----rw-rw- 66 fail.txt

`chmod 660 fail.txt` #sisuliselt ug=rw
-rw-rw---- 660 fail.txt

kaustad
kaustaõiguste vaatamiseks:
```
ll -d proov/
ls -ld proov/
stat -c '%A %a %n' proov/
```

`chmod 5 proov/` #sisuliselt 005 või o=rx
d------r-x 5 proov/

`chmod 75 proov/` #sisuliselt 075 või g=rwx,o=rx
d---rwxr-x 75 proov/

`chmod 755 proov/` #sisuliselt o=rwx,go=rx
drwxr-xr-x 755 proov/

Eriõigused
õiguste vaatamiseks:
```
stat -c '%A %a %n' <fail või kaust>
stat -c '%A %a %n' fail.txt
stat -c '%A %a %n' proov/
```

SetUID – kasutaja (omaniku) õigustes käivitamine, **omanikul peab olema käivitusõigus**
kaheksandsüsteemis: *chmod 4000*, tähtedega *chmod u+sx*

`chmod 4755 proov/` #SetUID määramine **kaustale**, lisaks määratakse 755 tavaõigustena
drw**s**r-xr-x 4755 proov/
… kui ei soovi, et grupp ega teised ligi saaks siis 700 on kataloogile tavaõigustena sobiv

`chmod 00755 proov/` #SetUID eemaldamiseks **kaustalt**, lisaks määratakse 755 tavaõigustena
drwxr-xr-x 755 proov/

õigused ei pärandu:
mkdir proov/test
> proov/test/proov.txt
stat -c'%A %a %n' proov/test/
drwxrwxr-x 775 proov/test/
stat -c'%A %a %n' proov/test/proov.txt
-rw-rw-r-- 664 proov/test/proov.txt

`chmod u+sx proov/` #SetUID lisamine **kaustale**, olemasolevaid õigusi ei muudeta
drws------ 4700 proov/

`chmod u-s proov/` #SetUID eemaldamine **kaustalt**, olemasolevaid õigusi ei muudeta
drwx------ 700 proov/

`chmod 4744 fail.txt` #SetUID määramine **failile**, lisaks määratakse 744 tavaõigustena (omanik peab saama käivitada SetUID puhul)

-rw**S**r--r-- 4744 fail.txt
… kui ei soovi, et grupp ega teised ligi saaks siis 700 on failile tavaõigustena sobiv koos SetUID'ga

`chmod 0644 fail.txt` #SetUID eemaldamine **faililt**, lisaks määratakse 644 tavaõigustena (sh eemaldatakse käivitusõigus)
-rw-r--r-- 644 fail.txt
… toimib ka *chmod 00644 fail.txt* kuigi otseselt vajalik ei ole

`chmod u+sx fail.txt` #SetUID lisamine **failile**, olemasolevaid õigusi ei muudeta
-rwsr--r-- 4744 fail.txt

`chmod u-s fail.txt` #SetUID eemaldamine **faililt**, olemasolevaid õigusi ei muudeta
-rwxr--r-- 744 fail.txt

SetGID – grupi õigustes käivitamine, **grupil peab olema käivitusõigus**
kaheksandsüsteemis: chmod 2000, tähtedega `chmod g+sx path`
ka loodavad alamkataloogid tulevad SetGID õigustega

`chmod 2755 proov/` #SetGID määramine **kaustale**, lisaks määratakse 755 tavaõigustena
drwxr-**s**r-x 2755 proov/

SetGID õigused päranduvad kataloogidele:
mkdir proov/test
stat -c'%A %a %n' proov/test/
drwxrw**s**r-x 2775 proov/test/
> proov/test/proov.txt
-rw-rw-r-- 664 proov/test/proov.txt

`chmod 00755 proov/` #SetGID eemaldamine **kaustalt**, lisaks määratakse 755 tavaõigustena
drwxr-xr-x 755 proov/

`chmod g+sx proov/` #SetGID määramine **kaustale**, olemasolevaid õigusi ei muudeta
drwxr-**s**r-x 2755 proov/

`chmod g-s proov/` #SetGID eemaldamine **kaustalt**, olemasolevaid õigusi ei muudeta
drwxr-xr-x 755 proov/

`chmod 2654 fail.txt` #SetGID määramine **failile**, lisaks määratakse 654 tavaõigustena
-rw-r-**s**r-- 2654 fail.txt

`chmod 0644 fail.txt` #SetGID eemaldamine **faililt**, lisaks määratakse 644 tavaõigustena (käivitusõigus grupile ei ole siis enam vajalik)
-rw-r--r-- 644 fail.txt

`chmod g+sx fail.txt` #SetGID lisamine **failile**, olemasolevad õigusi ei muudeta
-rw-r-**s**r-- 2654 fail.txt

`chmod g-sx fail.txt` #SetGID eemaldamine **faililt**, eemaldatakse käivitusõigus grupile, mis ei ole siis enam vajalik
-rw-r--r-- 644 fail.txt

kleepbitt (*sticky bit*) – vaid omanik saab muuta, kustutada; teistel peab olema käivitusõigus

kaheksandsüsteemis: chmod 1000, tähtedega `chmod +tx path`
üldjuhul kasutatakse siis kui on ühiskasutuses olevad kaustad kuid igaüks tohib ise omi faile hallata

vt näiteks:

`stat -c'%A %a %n' /tmp`

drwxrwxrwt 1777 /tmp

`stat -c'%A %a %U %u %G %g %n' /tmp/*`

%A õigused inimloetavalt
%a õigused kaheksandsüsteemis
%U kasutajanimi (omanik)
%u kasutaja ID
%G grupi nimi
%g grupi ID
%n faili, kausta nimi

`stat -c'%A %a %n' /var/tmp/`

drwxrwxrwt 1777 /var/tmp

`chmod 1777 proov/` #kleepbiti määramine **kaustale**, lisaks määratakse 777 tavaõigustena

drwxrwxrwt 1777 proov/

kleepbiti puhul õiguste pärandumist ei esine:

mkdir proov/test/

stat -c'%A %a %n' proov/test/

drwxrwxr-x 775 proov/test/

proov/test/proov.txt

stat -c'%A %a %n' proov/test/proov.txt

-rw-rw-r-- 664 proov/test/proov.txt

`chmod 0755 proov/` #kleepbiti eemaldamine **kaustalt** (seekord piisab ka ühest nullist), lisaks määratakse 755 tavaõigustena (muutmisõigus grupile, teistele ei ole enam vajalik)

drwxr-xr-x 755 proov/

`chmod 1755 fail.txt` #kleepbiti määramine **failile**, lisaks määratakse ka 755 tavaõigustena

-rwxr-xr-**t** 1755 fail.txt

`chmod 0644 fail.txt` #kleepbiti eemaldamine **faililt**, lisaks määratakse ka 644 tavaõigustena (käivitusõigus ei ole enam vajalik)

-rw-r--r-- 644 fail.txt

`chmod +tx` #kleepbiti määramine **failile**, lisaks määratakse ka 755 tavaõigustena

-rwxr-xr-**t** 1755 fail.txt

`chmod -tx` #kleepbiti eemaldamine **faililt**, lisaks määratakse ka 644 tavaõigustena (käivitusõigus ei ole enam vajalik)

-rw-r--r-- 644 fail.txt

Kataloogid on failid

Esialgu on ehk natuke kummaline mõelda kataloogidest kui erilistest failidest, milles on kirjas seal sisalduvate failide nimed, kuid tegelikult vastab see failisüsteemi tehnilisele korraldusele.

Niisiis, kataloog on fail, kus on kirjas selles kataloogis sisalduvate failide ja nn alamkataloogide nimed ning viited failide tegelikele asukohtadele.

Selgitame:

- Enne ütlesime, et kui meil pole kataloogi (st kataloogi kui faili) lugemisõigust (r), siis ei saa seal sees anda käsku ls. Tõepoolest, et ls saaks midagi näidata, peab ta saama faili (st kataloogi kui faili) lugeda. Lugeda aga ilma lugemise õiguseta ei saa.
- Kui pole antud kataloogi (ehk erilise faili) kirjutamise õigust (w), siis ei saa sinna teha muutusi - ei saa faile kustutada ega juurde teha.
- Ja kõige tähtsam, kui meil pole kataloogi (st kui faili) käivitamise õigust (x), siis ei saa sinna isegi sisse minna (käsuga cd).

Failide ja kataloogide õigustega manipuleerides saab luua esmapilgul kummalisi tingimusi:

- keelates kataloogi kirjutamise ja lubades selles kataloogis olevasse faili kirjutada, saab faili sisu muuta
- lubades kasutajaid oma kataloogi sisse ja sinna kirjutada, kuid mitte lubades kataloogi lugeda, saavad nad sinna faile luua ilma, et nad neid käsuga ls näeks.
- lubades teisel kasutajal teha Teie kataloogi faile, on teisel kasutajal võimalus sinna tekitada ka nö kataloog ning kui ta moodustab selle kataloogi sisse faile, siis ei saa te neid maha võtta, kui ta pole Teile selleks spetsiaalselt õigust andnud. Sel lihtsal põhjusel, et te pole kataloogi omanik, kus need failid asuvad; faili kustutamine eeldab aga kataloogi kirjutamise õigust.

umask

Määrab vaikimisi loodavate failide, kaustade õigused. Ei toeta eriõigusi (*SetUID, SetGID, sticky bit*).

https://www.cyberciti.biz/tips/understanding-linux-unix-umask-value-usage.html
http://wintelguy.com/umask-calc.pl – arvutamine
http://www.webune.com/forums/umask-calculator.html
https://en.wikipedia.org/wiki/Umask

tavaliselt on umask väärtusega 002

hetkel kehtiva umask'i vaatamine
umask
0077

umask -S #sümbolitega
u=rwx,g=,o=

probleem: vaikimisi tekivad failid grupile lugemisõigusega → umask'i teine number peab 0 olema

umask'i sätted:
süsteemilaiune /etc/profile
/etc/login.defs → vajalik muuta USERGROUPS_ENAB → no (vajalik kasutajaga uuesti sisselogimine) – siis hakkab seal failis määratud UMASK tööle.

failis */etc/pam.d/common-session* on määratud: *session optional pam_umask.so* (vajadusel paigaldada pam-modules), jäigalt saab sinna faili kirjutada *session optional pam_umask.so umask=002*
(jõustamiseks vajalik süsteemi taaskäivitus)

kasutaja ~/.profile (ka: ~/.bashrc); uutele kasutajatele: /etc/skel/.profile
superkasutaja võimalus muuta umask'i konkreetsele kasutajale:
`grep student /etc/passwd`
`student:x:1000:1000:student,,,:/home/student:/bin/bash` #algselt
sudo chfn --other='umask=022' student
`grep student /etc/passwd`
`student:x:1000:1000:student,,,,umask=022:/home/student:/bin/bash` #umask lisatud
sudo chfn --other='' student #taas umask'i eemaldamine

umask 000 -> 777(rwxrwxrwx)/666(rw-rw-rw-)
umask 002 -> 775(rwxrwxr-x)/664(rw-rw-r--)
umask 007 -> 770(rwxrwx---)/660(rw-rw----)
umask 022 -> 755(rwxr-xr-x)/644(rw-r--r--)
umask 027 -> 750(rwxr-x---)/640(rw-r-----)
umask 077 -> 700(rwx------)/600(rw-------)
umask 277 -> 500(r-x------)/400(r--------)

ainult grupp saab kõik õigused: umask 707
* kataloogiõigused: 777 – 707 = 070 (---rwx---)
* failiõigused: 666 – 707 = 060 (---rw----)

8nd-süsteemi väärtus : õigus
0 : lugemine, kirjutamine, käivitamine
1 : lugemine, kirjutamine,
2 : lugemine, käivitamine
3 : lugemine
4 : kirjutamine, käivitamine
5 : kirjutamine
6 : käivitamine
7 : õigused puuduvad

Esimesel kohal omanik, teisel grupp, kolmandal kõik ülejäänud.

Näiteks 077:

Bitt	Sihtgrupp	Õigus
0	omanik:	lugemine, kirjutamine, käivitamine
7	grupp:	õigused puuduvad
7	teised:	õigused puuduvad

Kataloogi puhul käivitusõigus (x) tähendab õigust sinna siseneda. Kataloogi sisu kuvamiseks on vajalik veel ka lugemisõigus (r):
ls -la
d**r**w**xr**-**xr**-**x** 2 student student 4096 Mar 18 23:48 kaust/
… antud juhul saab nii omanik

umask’i alusel õiguste arvutamine

vaikimisi kataloogid 777 ja failid 666 ning umask väärtus lahutatakse nendest ja saadakse reaalsed õigused. Negatiivse väärtuse korral kirjutatakse 0.

näiteks: umask 022
* kataloogiõigused: 777 – 022 = 755
* failiõigused: 666 – 022 = 644