

Filtrid

on hulk P-e, mille I-ks on 1 voog (F), millega tehakse mingid operatsioonid (tegevused) ja tulemus kirjutatakse 2.-e voogu (F)

grep, tail, sort, wc, ... - nim. filtriteks

Grep'ide perekond (grep, fgrep, egrep)

(global regular expression print)

\$ grep shabloon failid

väljastab iga rea, kus esineb shabloon; sõnade (stringide) otsimiseks kasut. enamikku standartsete tekstiredaktorite süntaktilistest võimalustest, *ed*

\$ grep file parimf.p

opfail = file of opilane;

\$ ls | grep .p

elli.p hulk.p kaup kaup.dat kaup.p mat.p matprot matprot.doc
matprot.p maxint.p nimi.p parim.p parimf.p rist.p temp topelt topelt.p
miks nii?

baseerub *ed* nn. regulaarsetel avaldistel (vt. *lex*'i reg. avaldised)

\$ ls | fgrep .p

elli.p hulk.p kaup.p mat.p matprot.p maxint.p nimi.p parim.p
parimf.p rist.p topelt.p

\$ ls -l | grep '^...x' - F-d, mida omanik saab täita

-rwxr-xr-x 1 avo 10240 Mar 12 04:49 a.out

egrep - shabloonid võivad olla keerulisemad, aeglasem kui *grep*

fgrep - shabloonid võivad olla ainult fikseeritud stringid, ei interpreteeri metasümboleid, kiirem

grep nii võimalustelt kui kiiruselt keskel

võtmed:

-n - väljastatakse ridade #-d

-v - tingimus vastupidine

```
$ ls | grep -v -n p
```

```
2:harjuta1      6:kool.dat      18:sis.dat      19:sis1.dat
```

```
20:sis2.dat
```

sort - sordib sisendvoo ridade kaupa vastavalt ASCII koodi-
tabelile; mõned võtmed:

-b - ignoreerib rea algul olevaid tühikuid ja tab'e

-d - sordib andmeid vastavalt sõnaraamatu järjestusele

-f - ei erista suur- ja väiketähti

-i - ignoreerib sümboleid väljaspool ASCII piirkonda 040-0176

-n - sordib numbreid sisaldavad read numbrite järgi

-r - sordib kahanevalt

-u - esitab ainult erinevad (unique)

+m - jätab m välja vahele

```
$ ls | sort -f - tähestiku järgi
```

```
$ ls -s | sort -n - pikkuse (plokkide arvu) järgi
```

```
$ ls -l | sort +3nr - pikkuse (baitide arvu)
```

uniq

uniq - eemaldab kõrvutiolevate korduvate ridade grupist kõik peale 1
olgu F 'avo':

```
Avo
```

```
avo
```

```
Avo
```

```
avo
```

```
$ uniq avo
```

```
Avo
```

```
avo
```

```
Avo
```

```
avo
```

```
$ sort -f avo | uniq
```

```
Avo
```

```
avo
```

comm

comm - võrdleb F-e

väljastab 3-e veergu:

1. read, mis on ainult f1-s

2. read, mis on ainult f2-s

3. read, mis on mõlemas

\$ comm -12 f1 f2 1. ja 2. veergu ei väljastata

\$ comm -3 f1 f2 3. veergu ei väljastata

tr (translate characters)

sümbolite transleerimiseks

\$ tr A-Z a-z < avo

avo

avo

avo

avo

\$ tr avo lea < avo

Aea

lea

Aea

lea

awk

sarnane vooredaktori *sed*'iga

awk - pattern scanning and processing language

\$ awk [-Fc] [-f prog] [-] [file...]

P loeb sisendvoogu ridade kaupa; ridu võrreldakse shabloonidega, mis on antud 'prog'-is; iga rea puhul, mis vastab shabloonile, täidetakse vastav tegevus

'prog' näeb välja järgmine:

shabloon {tegevus}

shabloon {tegevus}

...

‘prog’ võib olla ka käsuF

shabloonid on kas ‘sed’i regulaarsed avaldised (egrep) või

awk shabloonid

shabloon või tegevus võib puududa

kui puudub shabloon, siis ta vastab igale reale

kui puudub tegevus, siis väljastatakse vastav rida

sisendvoo iga rida jagatakse väljadeks (eraldajad probell(id) või

tab’id, võib ka määrata muud), *who* väljundvoo ridades on 6 välja

\$ who

a960440 tty11 May 5 07:18 (LAT_08002B035A12)

tamm tty12 May 5 11:50 (LAT_08002B035A12)

janm tty13 May 5 11:50 (LAT_08002B035A12)

a960759 ttyp8 May 5 09:58 (liisa.va.ttu.ee)

kyngas ttypc May 5 11:40 (class210_8.li.tt)

väljasid tähistatakse \$1, \$2, ... , \$NF, kus NF - väljade arv (praegu 6)

tahan 1. ja 5. välja, sortitud tähestiku järgi:

\$ who | awk '{print \$1, \$5}' | sort

a960440 07:18

a960759 09:58

janm 11:50

kyngas 11:40

tamm 11:50

\$ who | awk '{print NR, \$0}'

1 y94035 tty10 May 5 08:22 (LAT_08002B035A12)

2 a960440 tty11 May 5 07:18 (LAT_08002B035A12)

3 tamm tty12 May 5 11:50 (LAT_08002B035A12)

4 janm tty13 May 5 11:50 (LAT_08002B035A12)

5 margus tty14 May 5 11:58 (LAT_08002B0485FE)

NR - rea # \$0 - terve rida sisendvoos

shabloonid **BEGIN JA END**

BEGIN'i järel olevad tegevused täidetakse enne 1. rea sisestust
(algväärtustamine, päise väljastus, välja eraldaja määramine, ...)

END'i järel olevad tegevused täidetakse pärast viimase rea sisestust
\$ ls -l | awk 'END {printf "%d\n", NR}'

202

printf on sarnane analoogilise C funktsiooniga
aritmeetika ja muutujad

awk võimaldab teha rehkendusi sisendandmetega:

olgu meil vaja liita 1. veeru arvud, teeme käsuF 'kf':

```
{s=s+$1}
```

```
END {printf"%d\n", s}
```

```
$ wc * | awk -f kf
```

976

olgu 'kf1':

```
{s+= $1}
```

```
END {print s, s/NR}
```

```
$ wc * | awk -f kf1
```

980 89

muutujad võivad olla skaalarid, massiivide elemendid (x[i]) või väljad
muutujad initsialiseeritakse 0-ks, muutujaid ei ole vaja kirjeldada

awk-s võib kasutada järgmisi eritähendusega muutujaid:

NR, NF, FILENAME (jooksva IF'i nimi), FS (väljaeraldaja I-voos),
OFS (väljaeraldaja O-voos), ORS (reaeraldaja O-voos), OFMT (arvude
O-formaat, vaikimisi "%.6g")

lubatud tehted:

+, -, *, /, %

C tehted:

++, --, +=, -=, *=, /=, ja %=

!, ||, &&, ja sulud

regulaarsed avaldised peavad olema //-de vahel

tegevus on lausete jada

lubatud laused:

if (conditional) statement [else statement]

while (conditional) statement

for (expression ; conditional ; expression) statement

break

continue

variable = expression

print [expression-list] [>expression]

printf format [, expression-list] [>expression]

next # skip remaining patterns on this input line

exit # skip the rest of the input

eraldajateks on: ';' või uus rida

string konstant on "..."-s

functions exp, log, sqrt, and int

substr(s, m, n) tagastab stringi s n-sümbolist koosneva alamstringi

alates positsioonist m

näide massiivide kasutamise peale:

olgu F 'palk':

Jaan 50

Reet 100

Tom 200

Reet 30

Jaan 400

Tom 300

ja käsuF 'kf2':

{sum[\$1]+=\$2}

END {for (nimi in sum) print nimi " " sum[nimi]}

kasutatakse spetsiaalsed for-laused
for (muutuja in massiiv)

lause

\$ awk -f kf2 < palk

Tom 500

Jaan 450

Reet 130

mõned momendid:

- väljundvoo võib suunata *print* ja *printf* ga F-i (>) või torusse (|)
>> - tähendab lisamist
- shabloonide paari abil saab anda ridade diapasooni
NR ==10, NR ==20 määratakse ridade diapsoon 10..20 (incl.)