

Komanda awk

- Komanda **awk** se može koristiti za obradu sadržaja fajlova
- Fajl se obrađuje liniju po liniju po principu:
 - Skeniranje linije
 - Podela linije na polja
 - Poređenje linije/polja sa zadatim obrascem
 - Obavljanje zadate akcije
- Komanda **awk** može biti deo pajpa, ali može da prihvati kao argument 1 ili više fajlova
- Unutar awk se mogu koristiti regularni izrazi, operatori, konstrukcije poput if ili petlji
- Sintaksa prilično preuzeta iz C programskog jezika i ERE

Struktura komande

- Komanda **awk** se poziva po sledećem principu:
awk opcije 'izbor {akcija}' ulazni fajlovi
- Ako se rezultat želi ispisati u fajl umesto u terminal, može se koristiti redirekcija izlaza
- Najčešće korišćene opcije su -F, -v, -f
- Opcija -F definiše delimiter između polja (difolt je space i tab)
- Opcija -v se navodi u obliku -v *var=vrednost* i koristi se za podešavanje vrednosti varijable
- Opcija -f se navodi u obliku -f fajl i koristi se kad je u fajlu navedeno šta awk treba da uradi ('**izbor {akcija}**' deo)

Osnovi princip rada

- Ideja je da se fajl obrađuje liniju po liniju
- Svaka linija se deli na polja
- Poljima se pristupa kao \$1, \$2, ... gde \$i predstavlja *i*-to polje (polja se numerišu od 1)
- \$0 predstavlja čitavu liniju tzv. zapis (record)
- Da bi se definisalo koje linije se žele obraditi može se koristiti *izbor* deo komande, npr. regularni izraz
- Regularni izraz se navodi između // tj. /reg.izraz/
- Ako se *izbor* izostavi onda se podrazumeva obrada svih linija zadatom akcijom
- Ako se izostavi *akcija* onda se podrazumeva **print** akcija

Primer

```
ubuntu@ubuntu2004:~/term5$ cat kolone1| awk '{print}'
Studenti
Jovan Jovic 2015 0264
Mile Milic 2016 0357
Jovana Panic 2015 0145
Jelena Maric 2016 0023
Petar Popovic 2016 0043
Igor Kovic 2014 0434
Rade Dabic 2015 0333
ubuntu@ubuntu2004:~/term5$ cat kolone1| awk '/[[[:digit:]]]/'
Jovan Jovic 2015 0264
Mile Milic 2016 0357
Jovana Panic 2015 0145
Jelena Maric 2016 0023
Petar Popovic 2016 0043
Igor Kovic 2014 0434
Rade Dabic 2015 0333
```

```
ubuntu@ubuntu2004:~/term5$ cat kolone1| awk '/[[[:digit:]]/ {print $1,$2}'
Jovan Jovic
Mile Milic
Jovana Panic
Jelena Maric
Petar Popovic
Igor Kovic
Rade Dabic
ubuntu@ubuntu2004:~/term5$ cat kolone1| awk '/[[[:digit:]]/ {print $1 $2}'
JovanJovic
MileMilic
JovanaPanic
JelenaMaric
PetarPopovic
IgorKovic
RadeDabic
```

```
ubuntu@ubuntu2004:~/term5$ cat kolone1| awk '/[[[:digit:]]/ {print $2,$1}'
Jovic Jovan
Milic Mile
Panic Jovana
Maric Jelena
Popovic Petar
Kovic Igor
Dabic Rade
```

```
ubuntu@ubuntu2004:~/term5$ cat kolone1| awk '/[[[:digit:]]/ {print "Student " $0}'
Student Jovan Jovic 2015 0264
Student Mile Milic 2016 0357
Student Jovana Panic 2015 0145
Student Jelena Maric 2016 0023
Student Petar Popovic 2016 0043
Student Igor Kovic 2014 0434
Student Rade Dabic 2015 0333
```

Izbor

- Regularni izraz (pattern) se može koristiti za selekciju linija koje se žele obraditi
- BEGIN i END predstavljaju specijalne izbore
- BEGIN se uvek prvi procesiraju i oni idu na početak fajla – idealno za dodavanje željenog zaglavlja (header) dokumentu
- END se uvek poslednji procesiraju i ide na kraj fajla – idealno za dodavanje željenog začelja (footer) dokumentu ili za ispis rezultata obrade npr. prosečna vrednost polja iz zapisa i sl.

Izbor

- Mogu se koristiti i logički (&&, ||, !) i relacioni operatori (<, >, >=, <=, ==, !=)
- Logički su zgodni za povezivanje više patterna ili za negaciju patterna
- Može se formirati i opseg linija navođenjem pattern1, pattern2 gde pattern1 predstavlja početak opsega, a pattern2 kraj opsega
- Postoji čak i mogućnost upotrebe ?: operatora iz C jezika
pattern1 ? pattern2 : pattern3

Primer

```
ubuntu@ubuntu2004:~/term5$ cat kolone1 | awk 'BEGIN {print "ZAGLAVLJE"}'  
ZAGLAVLJE  
ubuntu@ubuntu2004:~/term5$ cat kolone1 | awk 'END {print "ZACELJE"}'  
ZACELJE  
ubuntu@ubuntu2004:~/term5$ cat kolone1 | awk 'BEGIN {print "ZAGLAVLJE"}  
> /[[:digit:]]/ {print $0}  
> END {print "ZACELJE"}'  
ZAGLAVLJE  
Jovan Jovic 2015 0264  
Mile Milic 2016 0357  
Jovana Panic 2015 0145  
Jelena Maric 2016 0023  
Petar Popovic 2016 0043  
Igor Kovic 2014 0434  
Rade Dabic 2015 0333  
ZACELJE
```

Primer

```
ubuntu@ubuntu2004:~/term5$ cat kolone1 | awk '$3==2016 {print}'
Mile Milic 2016 0357
Jelena Maric 2016 0023
Petar Popovic 2016 0043
ubuntu@ubuntu2004:~/term5$ cat kolone1 | awk '$3<2016 {print}'
Studenti
Jovan Jovic 2015 0264
Jovana Panic 2015 0145
Igor Kovic 2014 0434
Rade Dabic 2015 0333
ubuntu@ubuntu2004:~/term5$ cat kolone1 | awk '$3!=2016 && $4>200 {print}'
Jovan Jovic 2015 0264
Igor Kovic 2014 0434
Rade Dabic 2015 0333
ubuntu@ubuntu2004:~/term5$ cat kolone1 | awk '$3!=2016 && $4<200 {print}'
Studenti
Jovana Panic 2015 0145
ubuntu@ubuntu2004:~/term5$ cat kolone1 | awk '! ($3==2016) {print}'
Studenti
Jovan Jovic 2015 0264
Jovana Panic 2015 0145
Igor Kovic 2014 0434
Rade Dabic 2015 0333
```

Primer

```
ubuntu@ubuntu2004:~/term5$ cat kolone1 | awk '!/[[:digit:]]/ {print}'
Studenti
ubuntu@ubuntu2004:~/term5$ cat kolone1 | awk '/[[:digit:]]/ && /^J/ {print}'
Jovan Jovic 2015 0264
Jovana Panic 2015 0145
Jelena Maric 2016 0023
ubuntu@ubuntu2004:~/term5$ cat kolone1 | awk '/^J/ ? $3==2015 : $3==2016 {print}'
Jovan Jovic 2015 0264
Mile Milic 2016 0357
Jovana Panic 2015 0145
Petar Popovic 2016 0043
```

Built-in promenljive

- ARGV – broj argumenata prosleđen awk komandi
- ARGV – niz argumenata koji je prosleđen awk komandi
- ENVIRON – niz koji omogućava pristup promenljivama okruženja
- FILENAME – naziv fajla koji se trenutno obrađuje (nedefinisan po defaultu unutar BEGIN dela, mora da se zasebno setuje)
- FNR – redni broj linije trenutno obrađivanog fajla
- FS – delimiter polja
- NF – broj polja trenutnog zapisa (linije)
- NR – redni broj linije koja se trenutno obrađuje (sa stanovišta svih fajlova)

Built-in promenljive

- RS – delimiter zapisa (newline je default)
- OFS – izlazni delimiter polja (space je default)
- ORS – izlazni delimiter zapisa (newline je default)

Primer

```
ubuntu@ubuntu2004:~/term5$ awk '{print NR,$0}' kolone1
1 Studenti
2 Jovan Jovic 2015 0264
3 Mile Milic 2016 0357
4 Jovana Panic 2015 0145
5 Jelena Maric 2016 0023
6 Petar Popovic 2016 0043
7 Igor Kovic 2014 0434
8 Rade Dabic 2015 0333
ubuntu@ubuntu2004:~/term5$ awk '{print FNR,$0}' kolone1
1 Studenti
2 Jovan Jovic 2015 0264
3 Mile Milic 2016 0357
4 Jovana Panic 2015 0145
5 Jelena Maric 2016 0023
6 Petar Popovic 2016 0043
7 Igor Kovic 2014 0434
8 Rade Dabic 2015 0333
```

```
ubuntu@ubuntu2004:~/term5$ awk '{print FNR,$0}' kolone1 kolone1
1 Studenti
2 Jovan Jovic 2015 0264
3 Mile Milic 2016 0357
4 Jovana Panic 2015 0145
5 Jelena Maric 2016 0023
6 Petar Popovic 2016 0043
7 Igor Kovic 2014 0434
8 Rade Dabic 2015 0333
1 Studenti
2 Jovan Jovic 2015 0264
3 Mile Milic 2016 0357
4 Jovana Panic 2015 0145
5 Jelena Maric 2016 0023
6 Petar Popovic 2016 0043
7 Igor Kovic 2014 0434
8 Rade Dabic 2015 0333
ubuntu@ubuntu2004:~/term5$ awk '{print NR,$0}' kolone1 kolone1
1 Studenti
2 Jovan Jovic 2015 0264
3 Mile Milic 2016 0357
4 Jovana Panic 2015 0145
5 Jelena Maric 2016 0023
6 Petar Popovic 2016 0043
7 Igor Kovic 2014 0434
8 Rade Dabic 2015 0333
9 Studenti
10 Jovan Jovic 2015 0264
11 Mile Milic 2016 0357
12 Jovana Panic 2015 0145
13 Jelena Maric 2016 0023
14 Petar Popovic 2016 0043
15 Igor Kovic 2014 0434
16 Rade Dabic 2015 0333
```

Primer

```
ubuntu@ubuntu2004:~/term5$ awk '{print NF,$0}' kolone1
1 Studenti
4 Jovan Jovic 2015 0264
4 Mile Milic 2016 0357
4 Jovana Panic 2015 0145
4 Jelena Maric 2016 0023
4 Petar Popovic 2016 0043
4 Igor Kovic 2014 0434
4 Rade Dabic 2015 0333
ubuntu@ubuntu2004:~/term5$ awk 'BEGIN {print ENVIRON["HOME"]}'
/home/ubuntu
ubuntu@ubuntu2004:~/term5$ awk 'BEGIN {print ARGV, ARGV[0], ARGV[1]}' kolone1
2 awk kolone1
```

```
ubuntu@ubuntu2004:~/term5$ awk 'BEGIN {print FILENAME}' kolone1
ubuntu@ubuntu2004:~/term5$ awk 'BEGIN {getline; print FILENAME}' kolone1
kolone1
```

Operatori

<u>assignment</u>	= += -= *= /= %= ^=
<u>conditional</u>	? :
<u>logical or</u>	
<u>logical and</u>	&&
<u>array membership</u>	in
<u>matching</u>	~ !~
<u>relational</u>	< > <= >= == !=
<u>concatenation</u>	(no explicit operator)
<u>add ops</u>	+ -
<u>mul ops</u>	* / %
<u>unary</u>	+ -
<u>logical not</u>	!
<u>exponentiation</u>	^
<u>inc and dec</u>	++ -- (both post and pre)
<u>field</u>	\$

Konstrukcije

```
if ( expr ) statement
```

```
if ( expr ) statement else statement
```

```
while ( expr ) statement
```

```
do statement while ( expr )
```

```
for ( opt_expr ; opt_expr ; opt_expr ) statement
```

```
for ( var in array ) statement
```

```
continue
```

```
break
```

Primer

```
ubuntu@ubuntu2004:~/term5$ awk -F ';' -v sum=0,n=0 'BEGIN {print "Prosecna ocena"}
> {n++;sum+=$3}
> END {print sum/n}' kolone3
Prosecna ocena
8.14286
```

```
ubuntu@ubuntu2004:~/term5$ awk -F ';' 'BEGIN {print "Prosecna ocena"}
{n++;sum+=$3}
END {print sum/n}' kolone3
Prosecna ocena
8.14286
```

```
ubuntu@ubuntu2004:~/term5$ awk -F ';' 'BEGIN {print "Prosecna ocena"}
{n++;sum+=$3}
END {print sum/n}' kolone3
Prosecna ocena
8.14286
ubuntu@ubuntu2004:~/term5$ awk -F ';' -v sum=100,n=0 'BEGIN {print "Prosecna ocena"}
{n++;sum+=$3}
END {print sum/n}' kolone3
Prosecna ocena
22.4286
ubuntu@ubuntu2004:~/term5$ awk -F ';' -v n=0 'BEGIN {sum=100;print "Prosecna ocena"}
{n++;sum+=$3}
END {print sum/n}' kolone3
Prosecna ocena
22.4286
```

Nizovi

- Nizovi su asocijativni
- Članu niza se pristupa klasično navođenjem indeksa u uglastoj zagradi
- Za brisanje nekog člana se koristi *delete* (npr. `delete niz[0]`)

```
ubuntu@ubuntu2004:~/term5$ awk -F ';' 'BEGIN {print "Prosecna ocena"}
{n++;sum+=$3}
END {print sum/n}' kolone3
Prosecna ocena
8.14286
```

```
ubuntu@ubuntu2004:~/term5$ awk -F ';' -f awk_fajl kolone3
PROSECNA OCENA
8.14286
ubuntu@ubuntu2004:~/term5$ cat awk_fajl
BEGIN {n=0; sum=0;print "PROSECNA OCENA"}
{n++;sum+=$3}
END {print sum/n}
```

```
ubuntu@ubuntu2004:~/term5$ awk -F ';' -f awk_fajl kolone3
PROSECNA OCENA
8.14286
ubuntu@ubuntu2004:~/term5$ cat awk_fajl
BEGIN {n=0; sum=0;print "PROSECNA OCENA"}
{n++;sum+=$3}
END {print sum/n}
```

```
ubuntu@ubuntu2004:~/term5$ awk -F ';' -f awk_fajl2 kolone3
Ocena 10 ukupno 2
Ocena 8 ukupno 1
Ocena 6 ukupno 1
Ocena 5 ukupno 0
Ocena 9 ukupno 1
Ocena 7 ukupno 2
ubuntu@ubuntu2004:~/term5$ cat awk_fajl2
BEGIN {ocena[5]=0;ocena[6]=0;ocena[7]=0;ocena[8]=0;ocena[9]=0;ocena[10]=0;}
{ocena[$3]++}
END {for (i in ocena) print "Ocena " i " ukupno " ocena[i]}
```

```
ubuntu@ubuntu2004:~/term5$ awk -F ';' -f awk_fajl2a kolone3
Ocena 10 ukupno 2
Ocena 8 ukupno 1
Ocena 6 ukupno 1
Ocena 5 ukupno 0
Ocena 9 ukupno 1
Ocena 7 ukupno 2
ubuntu@ubuntu2004:~/term5$ cat awk_fajl2a
BEGIN {ocena["5"]=0;ocena["6"]=0;ocena["7"]=0;ocena["8"]=0;ocena["9"]=0;ocena["10"]=0;}
{ocena[$3]++}
END {for (i in ocena) print "Ocena " i " ukupno " ocena[i]}
```

```
ubuntu@ubuntu2004:~/term5$ awk -F ';' -f awk_fajl2b kolone3
Ocena 10 ukupno 2
Ocena 8 ukupno 1
Ocena 6 ukupno 1
Ocena 5 ukupno 0
Ocena 9 ukupno 1
Ocena 7 ukupno 2
ubuntu@ubuntu2004:~/term5$ cat awk_fajl2b
BEGIN {for(i=0;i<6;i++) ocena[i+5]=0}
{ocena[$3]++}
END {for (i in ocena) print "Ocena " i " ukupno " ocena[i]}
```

```
ubuntu@ubuntu2004:~/term5$ awk -F ';' -f awk_fajl3 kolone3
Broj 9 i 10 je 3
Broj ostalih ocena je 4
ubuntu@ubuntu2004:~/term5$ cat awk_fajl3
BEGIN {manje=0;vece=0;}
{if($3>8) vece++; else manje++}
END {print "Broj 9 i 10 je " vece "\n" "Broj ostalih ocena je " manje}
```

```
ubuntu@ubuntu2004:~/term5$ awk -F ';' -f awk_fajl3a kolone3
Vise je manjih ocena
ubuntu@ubuntu2004:~/term5$ cat awk_fajl3a
BEGIN {vece=0;manje=0;}
{if($3>8) {vece++;manje--;} else {manje++;vece--;}}
END {if (vece>manje) print "Vise je vecih ocena"; else {if (vece<manje) print "Vise je manjih ocena"; else print "Jednako";}}
```

```
ubuntu@ubuntu2004:~/term5$ awk -F ';' -f awk_fajl4 kolone3
Prosečna ocena je 8.14286
ubuntu@ubuntu2004:~/term5$ cat awk_fajl4
function prosek(niz, i,sum,n)
{
    sum=0;
    n=0;
    for(i in niz) {n++; sum+=niz[i]}
    return sum/n
}

{niz[NR]=$3}
END {x=prosek(niz); print "Prosečna ocena je " x}
```

Mogu se koristiti i funkcije.

I argumenti funkcije i lokalne promenljive idu u deklaraciju funkcije.

Lokalne promenljive se razdvajaju sa višestrukim uzastopnim space karakterima od argumenata funkcije.

Ugrađene funkcije

- Postoji velik broj funkcija koji je na raspolaganju
- Mogu se pronaći u uputstvu awk komande
- Aritmetičke: atan2(y, x), cos(x), exp(x), int(x), log(x), rand(), sin(x), sqrt(x), srand([x])
- String: gsub(regex, sub, string), index(str, sub), length(str), match(str, regex), split(str, arr, regex), sprintf, strtonum(str), sub(regex, sub, string), substr(str, start, l), tolower(str), toupper(str),
- Time: systime, mktime(datespec), strftime([format [, timestamp[, utc-flag]])
- Others: next, break, getline, nextfile, return, exit, system

```
ubuntu@ubuntu2004:~$ awk 'BEGIN {n=10; print log(n)}'  
2.30259  
ubuntu@ubuntu2004:~$ awk 'BEGIN {n=2.78; printf "exp je %f, floor je %d\n", exp(  
n), int(n)}'  
exp je 16.119021, floor je 2  
ubuntu@ubuntu2004:~$ awk 'BEGIN {print systime()}'  
1613003235  
ubuntu@ubuntu2004:~$ awk 'BEGIN {t=systime(); print strftime("%D %T", t, 1)}'  
02/11/21 00:29:08  
ubuntu@ubuntu2004:~$ awk 'BEGIN {for(i=0;i<3;i++) print rand()}'  
0.221288  
0.418585  
0.832549
```

```
ubuntu@ubuntu2004:~$ cat zapisi  
1378 1 25.01.2020.  
1256 2 21.01.2020.  
1555 1 29.01.2020.  
1444 2 30.01.2020.  
1555 5 01.02.2020.  
1555 1 27.01.2020.  
1256 4 29.01.2020.  
ubuntu@ubuntu2004:~$ awk '{if($2==1)next;print $0}' zapisi  
1256 2 21.01.2020.  
1444 2 30.01.2020.  
1555 5 01.02.2020.  
1256 4 29.01.2020.  
ubuntu@ubuntu2004:~$ awk '{if($2==1)getline;print $0}' zapisi  
1256 2 21.01.2020.  
1444 2 30.01.2020.  
1555 5 01.02.2020.  
1256 4 29.01.2020.
```

```
ubuntu@ubuntu2004:~$ cat zapisi
1378 1 25.01.2020.
1256 2 21.01.2020.
1555 1 29.01.2020.
1444 2 30.01.2020.
1555 5 01.02.2020.
1555 1 27.01.2020.
1256 4 29.01.2020.
ubuntu@ubuntu2004:~$ cat zapisi2
1378 2 25.01.2020.
1256 5 21.01.2020.
1555 3 29.01.2020.
1444 1 30.01.2020.
1555 5 01.02.2020.
1555 1 27.01.2020.
1256 4 29.01.2020.
ubuntu@ubuntu2004:~$ awk '{if($2==5)nextfile;print $0}' zapisi zapisi2
1378 1 25.01.2020.
1256 2 21.01.2020.
1555 1 29.01.2020.
1444 2 30.01.2020.
1378 2 25.01.2020.
ubuntu@ubuntu2004:~$ awk '{if($2==5)exit;print $0}' zapisi zapisi2
1378 1 25.01.2020.
1256 2 21.01.2020.
1555 1 29.01.2020.
1444 2 30.01.2020.
```

```
ubuntu@ubuntu2004:~$ cat zapisi
1378 1 25.01.2020.
1256 2 21.01.2020.
1555 1 29.01.2020.
1444 2 30.01.2020.
1555 5 01.02.2020.
1555 1 27.01.2020.
1256 4 29.01.2020.
ubuntu@ubuntu2004:~$ awk ' {i++; niz[i]=$1}
END {for (j in niz) print niz[j]}' zapisi
1555
1555
1555
1256
1378
1444
1256
ubuntu@ubuntu2004:~$ awk ' {i++; niz[i]=$1}
END {for (j in niz) {print niz[j]; if(niz[j]==1256) break}}' zapisi
1555
1555
1555
1256
```

```
ubuntu@ubuntu2004:~$ cat zapisi
1378 1 25.01.2020.
1256 2 21.01.2020.
1555 1 29.01.2020.
1444 2 30.01.2020.
1555 5 01.02.2020.
1555 1 27.01.2020.
1256 4 29.01.2020.
ubuntu@ubuntu2004:~$ awk '/^1555/ {print length($0); print substr($0, 1, 4)}' zapisi
18
1555
18
1555
18
1555
ubuntu@ubuntu2004:~$ awk '/^1555/ {print length($0) substr($0, 1, 4)}' zapisi
181555
181555
181555
ubuntu@ubuntu2004:~$ awk '/^1555/ {print length($0) " " substr($0, 1, 4)}' zapisi
18 1555
18 1555
18 1555
```

find

- Komanda za traženje fajlova kroz zadatu hijerarhiju direktorijuma
- Veoma moćna komanda sa velikim brojem opcija koje omogućavaju fino podešavanje kriterijuma pretrage
- U komandi je pored opcija koje podešavaju kriterijume pretrage potrebno navesti jednu ili više polaznih tačaka pretrage, a potom i izraz kojim se označava šta se želi pronaći i eventualno šta uraditi sa pronađenim fajlovima
- Opcije -P, -L i -H (jedna od njih) se navode na početku iza **find** (moraju se napisati pre prve polazne tačke)
- Opcija -P podrazumeva da se ne slede simbolički linkovi (ovo je difolt opcija i ne mora da se navede eksplicitno)
- Opcija -L podrazumeva da se slede simbolički linkovi ukoliko je to moguće
- Opcija -H podrazumeva da se ne slede simbolički linkovi sem u slučaju fajlova navedenih kao argument komande **find**

Lista polaznih tačaka pretrage

- Lista polaznih tačaka pretrage podrazumeva listu fajlova (suštinski direktorijuma) koji se koriste kao polazna mesta odakle pretraga kreće - ako se ne navede ništa u listi podrazumeva se trenutni radni direktorijum kao polazna tačka

```
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find dir11/dir21 dir12
dir11/dir21
dir11/dir21/bb.php
dir11/dir21/aa.txt
dir12
dir12/kf.pdf
```

Izraz

- Izraz predstavlja deo komande linije naveden iza liste polaznih tačaka pretrage
- Izraz se može sastojati iz testova, akcija, globalnih opcija, pozicionih opcija i operatora
- Testovi vraćaju true/false vrednost kao rezultat testa na nekom svojstvu fajla (na primer, da li je fajl prazan ili ne)
- Akcije imaju sporedne posledice (*side effects*) - npr. ispis na terminalu i takođe vraćaju true/false vrednost kao signalizaciju da li je akcija uspešno izvršena ili ne
- Globalne i pozicione opcije definišu način izvršenja testova/akcija tj. kriterijume pretrage - razlika ova dva tipa opcija je u širini važenja - globalne važe za kompletan izraz, a pozicione samo za deo izraza koji se nalazi iza njih
- Operatori omogućavaju povezivanje delova izraza u neku operaciju (npr. logički I ili ILL operator)

Globalne opcije

- Globalne opcije utiču na sve delove izraza, čak i ako postoje testovi i akcije navedeni pre njih
- Iz tog razloga je radi bolje razumljivosti i preglednosti bolje globalne opcije navesti na početku izraza, tj. neposredno iza liste polaznih tačaka pretrage
- -depth - procesira se prvo sadržaj direktorijuma pre samog direktorijuma (-d je alternativno pisanje ove komande, ali je preporuka da se ipak koristi -depth)
- -maxdepth L - definiše do koje dubine pretraga treba da ide (ako se za L stavi 0 onda se pretraga vrši samo u polazna tačka direktorijumu, ali ne i u poddirektorijumima)
- -mindepth L - definiše od koje dubine treba krenuti sa pretragom
- Kombinovanjem maxdepth i mindepth opcija se može definisati opseg dubina pretrage
- Postoje i druge globalne opcije (pogledati u man za **find**)

Globalne opcije - primer

Ako se samo navede **find** vrši se pretraga kompletnog radnog direktorijuma i njegovih poddirektorijuma. Isto važi i ako se navede **find .** pošto tačka označava radni direktorijum i time lista polaznih tačaka sadrži samo radni direktorijum.

Uočiti razliku u ispisu sadržaja kada se koristi **-depth** opcija. Direktorijumi uvek idu iza svojih sadržaja u listingu, za razliku od prvog primera gde navedena opcija nije korišćena.

```
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find
.
./pp.avi
./xx.txt
./dir11
./dir11/nn.txt
./dir11/dir22
./dir11/dir22/fff.pdf
./dir11/dir22/ccc.txt
./dir11/dir21
./dir11/dir21/bb.php
./dir11/dir21/aa.txt
./dir11/aa.jpg
./dir12
./dir12/kf.pdf
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find -depth
./pp.avi
./xx.txt
./dir11/nn.txt
./dir11/dir22/fff.pdf
./dir11/dir22/ccc.txt
./dir11/dir22
./dir11/dir21/bb.php
./dir11/dir21/aa.txt
./dir11/dir21
./dir11/aa.jpg
./dir11
./dir12/kf.pdf
./dir12
.
```

Globalne opcije - primer

Uočiti šta je nivo dubine 1, odnosno 2. Sam radni direktorijum je nivo 0. Fajlovi u njemu (obični i direktorijumi) su nivo 1. Fajlovi u direktorijumima nivoa 1 su nivo2, itd.

```
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find -mindepth 1
./pp.avi
./xx.txt
./dir11
./dir11/nn.txt
./dir11/dir22
./dir11/dir22/fff.pdf
./dir11/dir22/ccc.txt
./dir11/dir21
./dir11/dir21/bb.php
./dir11/dir21/aa.txt
./dir11/aa.jpg
./dir12
./dir12/kf.pdf
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find -mindepth 2
./dir11/nn.txt
./dir11/dir22
./dir11/dir22/fff.pdf
./dir11/dir22/ccc.txt
./dir11/dir21
./dir11/dir21/bb.php
./dir11/dir21/aa.txt
./dir11/aa.jpg
./dir12/kf.pdf
```

Globalne opcije - primer

Kombinovanjem opcija maxdepth i mindepth možemo kontrolisati opseg nivoa koji želimo pretražiti. Naravno, ove opcije se mogu kombinovati sa depth opcijom, što ilustruje poslednji primer u nizu.

```
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find -maxdepth 0
.
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find -maxdepth 1
.
./pp.avi
./xx.txt
./dir11
./dir12
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find -mindepth 1 -maxdepth 1
./pp.avi
./xx.txt
./dir11
./dir12
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find -mindepth 2 -maxdepth 2
./dir11/nn.txt
./dir11/dir22
./dir11/dir21
./dir11/aa.jpg
./dir12/kf.pdf
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find -mindepth 1 -maxdepth 2
./pp.avi
./xx.txt
./dir11
./dir11/nn.txt
./dir11/dir22
./dir11/dir21
./dir11/aa.jpg
./dir12
./dir12/kf.pdf
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find -depth -maxdepth 1
./pp.avi
./xx.txt
./dir11
./dir12
.
```

Testovi

- Testovi omogućavaju definisanje kriterijuma pretrage
- Prilikom pretrage fajlova, svaki fajl koji se nalazi na lokaciji koja je pretraživana se ispituje kroz zadate testove (ako ima zadatih testova) i gleda se da li zadovoljava test ili ne (u prevodu da li ispunjava kriterijum pretrage ili ne)
- Za neke testove se može koristiti referentni fajl (fajl koji korisnik sam navede) - npr. kriterijum može biti svi fajlovi noviji od referentnog fajla i sl.
- Ako se u nekim testovima koriste numeričke vrednosti, one se mogu navoditi u obliku n (vrednost mora biti tačno n), $+n$ (vrednost mora biti veća od n) i $-n$ (vrednost mora biti manja od n)

Testovi

- -amin n - test da li je fajlu pristupljeno pre tačno n minuta (ako se stavi $+n$ onda pre više od n minuta, a ako se stavi $-n$ onda pre manje od n minuta - u narednim testovima, gde god se spominje parametar n važi ista ova napomena)
- -anewer *ref_fajl* - test da li je fajlu pristupljeno nakon vremena modifikacije referentnog fajla *ref_fajl*
- -atime n - test da li je fajlu pristupljeno pre $n*24$ časova (u testu se u vremenu poslednjeg pristupa testiranog fajla frakcioni deo ne uzima u obzir, npr., ako je poslednji pristup bio pre $1.23*24h$ uzima se $1*24h$, uvek se gledaju celi brojevi)
- Testovi -cmin n , -cnewer *ref_fajl* i -ctime n rade isto što i prethodna tri testa, ali se sada gleda vreme promene statusa testiranog fajla, a ne vreme pristupa

Testovi

- Testovi `-mmin n`, `-newer ref_fajl` i `-mtime n` rade isto što i prethodni slični testovi, ali se sada gleda vreme promene sadržaja testiranog fajla
- `-empty` - test da li je fajl prazan
- `-executable` - test da li je fajl izvršni (tu spadaju i direktorijumi koji se mogu brauzovati) - postoje situacije u kojima ovaj test nije u potpunosti tačan
- `-nogroup` - test da numerički ID grupe fajla ne odgovara nijednoj grupi
- `-nouser` - test da numerički ID korisnika fajla ne odgovara nijednom korisniku

Testovi

- -name *pattern* - pretraga za fajlovima čija baza imena odgovara zadatom patternu (test sada podrazumeva poređenje naziva fajla sa patternom)
- Pod bazom imena se podrazumeva putanja fajla gde su skinuti vodeći direktorijumi
- Otuda u patternu ne bi trebalo da egzistira / jer nijedan fajl ne bi odgovarao takvom patternu
- Preporuka je da se pattern stavi pod navodnike da ne bi bio modifikovan u procesu *shell expansion*
- -path *pattern* - slično -name pretrazi, ali se sada gleda čitava putanja do fajla počev od neke od polaznih tačaka pretrage (iz tog razloga se u patternu karakteri . i / ne tretiraju kao specijalni)

Testovi

- -readable - test da li fajl može da se čita
- -writable- test da li fajl može da se piše
- -size n - veličina fajla je jednaka n - uz n se može navesti jedinica (b - 512-bajtni blokovi (b je difolt opcija), c - bajtovi, w - 16-bitne reči, k - kilobajti, M - megabajti, G - gigabajti)
- -type *tip* - test da li je fajl tipa *tip* (*tip* može biti b (*block*), c (*character*), d (direktorijum), p (*named pipe*), f (regularan fajl), l (simbolički link), s (soket))
- Može se navesti više tipova odjednom, i tada se oni razdvajaju zarezom prilikom navođenja
- uid n - test da li se numerički ID korisnika fajla poklapa sa n
- gid n - test da li se numerički ID grupe fajla poklapa sa n

Testovi

- -group *naziv_grupe* - da li fajl pripada navedenoj grupi
- -user *naziv_korisnika* - da li fajl pripada navedenom korisniku
- -regex *pattern* - da li se naziv fajla poklapa sa zadatim regularnim izrazom - traži se poklapanje na celoj putanji
- -perm *-biti* - test da li su svi navedeni biti dozvole setovani
- -perm */biti* - test da li je bilo koji od navedenih bita dozvole setovan
- -perm *biti* - test da li su dozvole dodeljene baš kako je navedeno u bitima
- Za bite se mogu koristiti simboličke oznake *u*, *g*, *o* (tri grupe dozvola - korisnik, grupa, svi ostali) ili se koristi oktalno označavanje (tri oktalne cifre, svaka predstavlja jednu grupu dozvola)

Testovi

- Neki od navedenih testova podržavaju i case insensitive varijantu
- Te varijante se navode sa slovom *i* na početku naziva testa
- Na primer, -iname, -ipath, -iregex

Akcije

- Akcije definišu akcije koje se izvršavaju na fajlovima
- -delete - brisanje fajlova
- -exec *komanda* - izvršava se navedena komanda - komanda se tumači do kraja linije ili do ;
- -exec *komanda* {} - izvršava se komanda na selektovanim fajlovima, gde {} predstavlja fajl koji je nađen **find** komandom
- -ls - listanje sadržaja fajla (kao da je pozvana ls -l)
- -ok komanda - isto što i -exec ali se prvo traži dozvola od korisnika
- -printf *format* - štampanje željenih podataka o nađenim fajlovima u zadatom formatu *format* na stdout izlaz (detalji o opcijama formatiranja se mogu naći u man uputstvu za **find**)

Akcije

- -print - štampanje punog naziva fajla na stdout izlaz
- -print0 - razlika u odnosu na print je što se nakon naziva fajla ne štampa *newline* karakter već null karakter
- Postoje i fprint verzije print komandi (fprintf, fprint i fprint0) koje ispis vrše u fajl, a ne stdout tj. terminal
- Kod fprint verzija mora da se navede fajl u koji će biti izvršen ispis (ako navedeni fajl ne postoji, biće kreiran)
- -prune - sprečava se ulazak u direktorijum (ako je fajl direktorijum)
- -quit - bezuslovni izlazak

Neke od opcija za formatiranje

- %f – naziv fajla
- %h – naziv direktorijuma
- %c – vreme promene statusa fajla
- %d – nivo u kom se fajl nalazi – gledano od početne tačke
- %i – inode broj
- %a – vreme pristupa
- %t – vreme modifikacije
- %s – veličina fajla u bajtovima
- %m – dozvole nad fajlom
- %u i %g – user i group vlasnik fajla
- %C, %T i %A mogu ići sa formatima vremena (D i F za datume, Y,m,d za godinu, mesec dan, H i M za sat i minut)

Pozicione opcije

- Ove opcije važe samo od mesta gde su navedene pa do kraja linije
- `-regextype` *tip* - podešavanje tipa regularnog izraza
- `-daystart` - merenje vremena od početka tekućeg dana, a ne od 24h ranije

```
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find -regextype help
find: Unknown regular expression type `help'; valid types are `findutils-default',
`awk', `egrep', `ed', `emacs', `gnu-awk', `grep', `posix-awk', `posix-basic', `po
six-egrep', `posix-extended', `posix-minimal-basic', `sed'.
```

Prikaz podržanih tipova regularnih izraza.

Operatori

- Delovi izraza se moraju povezati operatorom
- Ako se operator ne navede onda se po difoltu podrazumeva logičko I
- Logičko I se inače navodi sa -a ili -and
- Logičko I|I se navodi sa -o ili -or
- ! Ispred izraza označava negaciju izraza (može i -not, ali poželjnije je koristiti !)
- Zgrade obezbeđuju prioritet pa se njima može odrediti redosled izvršavanja izraza (iza otvarajuće i ispred zatvarajuće zgrade treba staviti space karakter)
- Zarez - funkcija liste - svi izrazi razdvojeni zarezom se uzimaju u obzir - pogodno kada se želi naći više različitih stvari, ali da se samo jednom prođe kroz hijerarhiju pretraživanih direktorijuma

Primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -empty
./xx.txt
./dir11/nn.txt
./dir11/dir22/fff.pdf
./dir11/dir21/bb.php
./dir11/dir21/dir31
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -nogroup
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -user ubuntu
.
./pp.avi
./xx.txt
./dir11
./dir11/nn.txt
./dir11/dir22
./dir11/dir22/fff.pdf
./dir11/dir22/ccc.txt
./dir11/dir21
./dir11/dir21/bb.php
./dir11/dir21/dir31
./dir11/dir21/aa.txt
./dir11/aa.jpg
./dir12
./dir12/kf.pdf
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -user ubuntu -name '*.txt'
./xx.txt
./dir11/nn.txt
./dir11/dir22/ccc.txt
./dir11/dir21/aa.txt
```

Primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -user ubuntu -name '*.txt'
./xx.txt
./dir11/nn.txt
./dir11/dir22/ccc.txt
./dir11/dir21/aa.txt
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -user ubuntu '!' -name '*.txt'
.
./pp.avi
./dir11
./dir11/dir22
./dir11/dir22/fff.pdf
./dir11/dir21
./dir11/dir21/bb.php
./dir11/dir21/dir31
./dir11/aa.jpg
./dir12
./dir12/kf.pdf
```

Primer upotrebe operatora negacije. Najbolje je staviti ! pod navodnike da se izbegne eventualno pogrešno tumačenje ovog karaktera. Ovde se ! mogao staviti i bez navodnika.

```
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -name '*.txt' -o -name '*.pdf'
./xx.txt
./dir11/nn.txt
./dir11/dir22/fff.pdf
./dir11/dir22/ccc.txt
./dir11/dir21/aa.txt
./dir12/kf.pdf
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -name '*.txt' -a -name '*.pdf'
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$
```

Primer upotrebe -o operatora. U drugom primeru je korišćen -a operator pa je rezultat prazan.

Primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -name '*.txt'
./xx.txt
./dir11/nn.txt
./dir11/dir22/ccc.txt
./dir11/dir21/aa.txt
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -maxdepth 2 -name '*.txt'
./xx.txt
./dir11/nn.txt
```

```
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -type d
.
./dir11
./dir11/dir22
./dir11/dir21
./dir11/dir21/dir31
./dir12
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -type f
./pp.avi
./xx.txt
./dir11/nn.txt
./dir11/dir22/fff.pdf
./dir11/dir22/ccc.txt
./dir11/dir21/bb.php
./dir11/dir21/aa.txt
./dir11/aa.jpg
./dir12/kf.pdf
```

Primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -path '.*dir.*'  
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -path '*dir*'  
./dir11  
./dir11/nn.txt  
./dir11/dir22  
./dir11/dir22/fff.pdf  
./dir11/dir22/ccc.txt  
./dir11/dir21  
./dir11/dir21/bb.php  
./dir11/dir21/dir31  
./dir11/dir21/aa.txt  
./dir11/aa.jpg  
./dir12  
./dir12/kf.pdf  
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -name '*dir*'  
./dir11  
./dir11/dir22  
./dir11/dir21  
./dir11/dir21/dir31  
./dir12
```

Primetiti razliku između -name i -path testa. Dodatno, tačka se ne tretira posebno u -path i -name testovima. Zato prvi primer nema rezultata pretrage.

Primeri

```
ubuntu@ubuntu-VirtualBox:~$ find . -perm 777
./dir1/slinkslinkfajl1
./dir1/slinkfajl1
ubuntu@ubuntu-VirtualBox:~$ find . -perm -o=w
./dir1/slinkslinkfajl1
./dir1/slinkfajl1
./ispis.txt
ubuntu@ubuntu-VirtualBox:~$ find . -perm /o=w,u=w,g=w -name '*.pdf'
./Documents/dir0/dir11/dir22/fff.pdf
./Documents/dir0/dir12/kf.pdf
./Downloads/intro-linux.pdf
ubuntu@ubuntu-VirtualBox:~$ find . -perm /222 -name '*.pdf'
./Documents/dir0/dir11/dir22/fff.pdf
./Documents/dir0/dir12/kf.pdf
./Downloads/intro-linux.pdf
ubuntu@ubuntu-VirtualBox:~$ find . -perm /002 -name '*.pdf'
```

```
ubuntu@ubuntu-VirtualBox:~$ find . -perm -o=w -name '*.pdf'
ubuntu@ubuntu-VirtualBox:~$ find . -perm -u=w,o=w -name '*.pdf'
ubuntu@ubuntu-VirtualBox:~$ find . -perm -u=w -name '*.pdf'
./Documents/dir0/dir11/dir22/fff.pdf
./Documents/dir0/dir12/kf.pdf
./Downloads/intro-linux.pdf
```

Primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -regex '.*dir2.*'  
./dir11/dir22  
./dir11/dir22/fff.pdf  
./dir11/dir22/ccc.txt  
./dir11/dir21  
./dir11/dir21/bb.php  
./dir11/dir21/dir31  
./dir11/dir21/aa.txt  
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -regex '*dir2*'
```

U slučaju -regex varijante, tačke se moraju koristiti. Drugi primer gde tačke nisu korišćene dao je prazan rezultat pretrage.

```
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -executable  
.  
./dir11  
./dir11/dir22  
./dir11/dir21  
./dir11/dir21/dir31  
./dir12
```

Primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -path '*dir1*' -prune
./dir11
./dir12
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -path '*dir1*'
./dir11
./dir11/nn.txt
./dir11/dir22
./dir11/dir22/fff.pdf
./dir11/dir22/ccc.txt
./dir11/dir21
./dir11/dir21/bb.php
./dir11/dir21/dir31
./dir11/dir21/aa.txt
./dir12
./dir12/kf.pdf
```

Primer upotrebe -prune opcije. Na linku <http://www.theunixschool.com/2012/07/find-command-15-examples-to-exclude.html> se mogu naći primeri primene prune opcije.

Primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -name '*.jpg'
./dir11/aa.jpg
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -name '*.jpg' -exec rm '{}' +
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -name '*.jpg'
```

```
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -name '*.pdf'
./dir11/dir22/fff.pdf
./dir12/kf.pdf
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -name '*.pdf' -exec rm -i '{}' \;
rm: remove regular empty file './dir11/dir22/fff.pdf'? n
rm: remove regular file './dir12/kf.pdf'? n
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -name '*.pdf' -exec rm -i '{}' +
rm: remove regular empty file './dir11/dir22/fff.pdf'? n
rm: remove regular file './dir12/kf.pdf'? n
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -name '*.pdf' -exec rm -i '{}' ;
find: missing argument to '-exec'
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -name '*.pdf' -ok rm -i '{}' \;
< rm ... ./dir11/dir22/fff.pdf > ? n
< rm ... ./dir12/kf.pdf > ? n
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -name '*.pdf' -ok rm -i '{}' +
find: missing argument to '-ok'
```

Ako se na kraju stavlja ; treba postaviti escape karakter \ ispred ;.

{ } predstavlja ono što je nađeno **find** komandom. Razlika između varijante sa + i sa ; je što + varijanta nađene fajlove stavlja kao jedan argument komande, a ; zasebno. Varijanta -ok radi samo sa ;

Primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find -name '*.txt' -printf '%p %s \n'

./xx.txt 0
./dir11/nn.txt 0
./dir11/dir22/ccc.txt 39
./dir11/dir21/aa.txt 37
./dir20/ff.txt 0
./dir20/fr.txt 0
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find -name '*.txt' -print
./xx.txt
./dir11/nn.txt
./dir11/dir22/ccc.txt
./dir11/dir21/aa.txt
./dir20/ff.txt
./dir20/fr.txt
```

Upotreba print komandi za ispis rezultata pretrage u terminalu. U prvom primeru opcija %p definiše naziv fajla, a opcija %s veličinu fajla u bajtovima. Opcija \n vrši prelazak u novu liniju čime se sprečava da podaci sledećeg nađenog fajla završe u istoj liniji. Pregled opcija formatiranja se može naći u man uputstvu za **find** komandu.

Primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -name '*ff*' \( -name '*.txt' -o -name '*.pdf' \)
./dir11/dir22/fff.pdf
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . \( -name '*.txt' -o -name '*.pdf' \) \
> -name '*ff*'
./dir11/dir22/fff.pdf
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -name '*ff*' \( -name '*.txt' -o -name '*.pdf' \)
find: paths must precede expression: (-name
Usage: find [-H] [-L] [-P] [-Olevel] [-D help|tree|search|stat|rates|opt|exec] [path...]
[expression]
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$
```

Primer upotrebe zagrada za formiranje logičkih izraza kod kojih je redosled bitan. Primiti da iza zagrade i ispred zagrade ide razmak, u suprotnom dolazi do greške kao što se vidi u poslednjem primeru. Setite se da shell razbija komandu na tokene i da je delimiter space karakter.

Primeri

```
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -name '*.txt' , -name '*.pdf'  
./dir11/dir22/fff.pdf  
./dir12/kf.pdf
```

U slučaju upotrebe zarezka kao operatora biće ispisan rezultat samo za poslednjeg člana liste.

```
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ find . -name '*.txt' -fprint ../fajl1 , -name  
'*.pdf' -fprint ../fajl2  
ubuntu@ubuntu-VirtualBox:~/Documents/dir0$ cd ..  
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl1  
./xx.txt  
./dir11/nn.txt  
./dir11/dir22/ccc.txt  
./dir11/dir21/aa.txt  
ubuntu@ubuntu-VirtualBox:~/Documents$ cat fajl2  
./dir11/dir22/fff.pdf  
./dir12/kf.pdf
```

Upotreba `fprint` akcije za svakog člana liste ponaosob će ispisati rezultate pretrage u navedene fajlove. Može se navesti i isti fajl za članove liste, ali to neće biti preterano korisno jer će ispis sledećeg člana u listi da prebriše deo ili sve od ispisa prethodnih članova liste.