**IV. Termin**

1. **I/O preusmeravanja u Unix/Linux-u**
2. **Zadatak**

Za pripremu za sledeće zadatke prekopirati folder **/myfilms** u svoj početni direktorijum!

[student@drvo1 ~]$ **cp –r /myfilms /home/student**

[student@drvo1 ~]$ **cd /home/student/myfilms**

Opcija *-r* omogućuje da se prekopira ceo direktorijum (zajedno sa fajlovima), bez ove opcije bi se dobila poruka greške.

***Standardni fajlovi dodeljeni procesima***

Sistem Unix/Linux je između ostalog i sistem orijentisan na fajlove, to se pokazuje prilikom izvršavanja naredbi: svakoj pokrenutoj naredbi se dodeljuju tri fajla:

* standardni input (stdin) (0)
* standardni output (stdout) (1)
* standardni error (stderr) (2)

Podrazumevano dodeljivanje ovih fajlova je sledeće: standardni input je dodeljen tastaturi terminala, dok su standardni output i error dodeljeni ekranu. Jedan tipičan program se izvršava na sledeći način: čita ulaz sa standardnog ulaza, iz ovoga produkuje neki rezultat koji prosleđuje na standarni izlaz, a eventualne poruke greške piše u standardni error fajl. Pomoću shell-a lako možemo promeniti standardne fajlove dodeljivanja.

1. **Zadatak**

Napraviti tekstualni fajl, odnosno listu fajlova o filmovima! Preći u prethodno prekopirani direktorijum *myfilms* gde svaki film ima redne brojeve, naslov i godinu proizvodnje u nazivu .html fajla! Izlistati sadržaj direktorijuma ali izlaz ne treba da se vidi na ekranu nego da se upiše u fajl myfilms.txt!

[student@drvo1 myfilms]$ **ls >myfilms.txt**

[student@drvo1 myfilms]$ **cat myfilms.txt**

001 - Baraka (1992).htm

002 - Reservoir Dogs (1992).htm

...

Da je već postojao fajl myfilms.txt njegov stari sadržaj bi se izgubio. Naredba *ls* umesto da je izlistao sadržaj direktorijuma na ekran, upisao ga je u fajl.

1. **Zadatak**

Ponovo izlistati sadržaj direktorijuma, sada se izlaz naredbe dodaje na kraj već postojećeg fajla myfilms.txt!

[student@drvo1 myfilms]$ **ls >> myfilms.txt**

***Povezivanje programa***

Za rešenje takvih problema je izmišljen **pipe** (čitaj: pajp, prevod: cevovod), koji povezuje standardni izlaz programa na levoj strani pipe-a sa standardnim ulazom programa na desnoj strani istog.

1. **Zadatak**

Izlistati sadržaj direktorijuma *myfilms* ekran po ekran, odnosno povezati izlaz naredbe *ls* na ulaz naredbe *more*!

[student@drvo1 myfilms]$ **ls -l | more**

Velika razlika u odnosu na prethodno rešenje je da u ovom slučaju shell istovremeno pokreće oba programa. U prethodnom slučaju je shell sačekao završetak prvog programa pre pokretanja drugog, sada u toku izvršavanja naredbe *ls* čim se pojavi neki izlaz, *more* odmah dobije te podatke i kreće u akciju. U vezi sa naredbom *pipe* treba spomenuti još jednu korisnu naredbu (ova naredba spada u sledeće poglavlje kod filtera): *tee*. Pomoću ove naredbe se postiže razgraničenje standardnog izlaza: naredba *tee* kopira svoj standardni input na standardni output i u fajl zadat parametrom. Ova naredba je najkorisnija u slučaju da se želi pratiti rad programa i radnje u unutrašnjosti pipe-a.

1. **Filteri (filters)**

Većina naredbi Unix/Linux-a (slično komandi *more*) čita svoj standardni input (najčešće po redovima), sa ulazom "radi nešto", zatim input, možda deo inputa ili neku vrednost koja zavisi od ulaza ispisuje na standardni izlaz. Ovu grupu naredbi nazivamo filterima (filters). Osobina filtara je da fajl zadat u prvom parametru uzimaju za standardni input.

1. **Zadatak**

Izlistati sadržaj direktorijuma */home/student/myfilms* i upisati u fajl *myfilms.txt*. Zatim izlistati sadržaj fajla pomoću naredbe *more* na dva načina: preusmeriti standardni ulaz i fajl predati naredbi kao parametar. Šta je razlika?

[student@drvo1 myfilms]$ **ls >myfilms.txt**

[student@drvo1 myfilms]$ **more <myfilms.txt**

[student@drvo1 myfilms]$ **more myfilms.txt**

Naredbe *more<myfilms.txt* i *more myfilms.txt* su u suštini iste: razlika je to da u prvom slučaju naredba *more* nema nijedan argument i preusmeravanje ulaza vrši shell, a u drugom slučaju *more* zna da je pokrenut sa argumentom i sam čita ulaz iz datog fajla. U narednim primerima sledi prikaz nekih korisnih filtara. Zajednička osobina svih filtera je da su svi samostalni mali pomoćni programi ali pomoću shell-a od njih se mogu napraviti programi za rešavanje komplikovanijih zadataka.

1. **Zadatak**

Pregled filtara u Unix/Linux-u.

* Najjednostavniji filter je *cat* koji svoj standardni ulaz red po red kopira na standardni izlaz. Npr:

[student@drvo1 myfilms]$ **cat >proba**

Ovo je proba.

Drugi red probnog fajla.

^D

Karakter u poslednjem redu je Ctrl+D koji za shell znači kraj fajla (slično karakteru Ctrl+Z u DOS-u).

Sadržaj novonastalog fajla se lista pomoću naredbe cat:

[student@drvo1 myfilms]$ **cat proba**

Ovo je proba.

Drugi red probnog fajla.

Napomena: isti rezultat daje i naredba *cat <proba*.

* Naredba *wc* (word count) broji redove, reči i karaktere koji dolaze sa inputa i ispisuje te brojeve:

[student@drvo1 myfilms]$ **cat proba | wc**

2 7 39

Isti rezultat daje *wc proba*.

* Jedan od najkorisnijih filtara je *grep* sa kojim možemo ispisati na izlaz one redove dobijenog teksta sa ulaza u kojima se javlja zadati uzorak:

[student@drvo1 myfilms]$ **cat proba | grep nog**

Drugi red probnog fajla.

1. **Zadatak**

Izlistati sve filmove iz direktorijuma *myfilms* koji su napravljeni 1994. godine! Koristiti filtar *grep*!

[student@drvo1 myfilms]$ **ls | grep 1994**

013 - Natural Born Killers (1994).htm

045 - Trzy kolory Bialy (1994).htm

046 - Trois couleurs Rouge (1994).htm

052 - Pulp Fiction (1994).htm

074 - Forrest Gump (1994).htm

* Često se koristi i filter *sort* koji redove sa standardnog ulaza ređa po nekom kriterijumu i tako ih prosleđuje na izlaz. Ako je naredba pozvana bez opcija, poređenje će biti po abecedi, a zadavanjem opcije *-r* rezultat će biti poređan u obrnutom redosledu.

1. **Zadatak**

Kreirati fajl *myfilms\_revers.txt* na osnovu fajla *myfilms.txt* tako da filmovi budu u obrnutom redosledu!

[student@drvo1 myfilms]$ **sort -r myfilms.txt >myfilms-revers.tx**t

Naredba *sort* može da sortira i numeričke podatke ili da sortira po poljima redova; proveriti mogućnosti naredbom *man*.

* Naredba *cut* se koristi za sečenje horizontalnih delova iz redova inputa. Npr. ako treba da se vide samo karakteri 3.-9. svakog reda:

[student@drvo1 myfilms]$ **cut -c3-9 proba**

o je pr

ugi red

Ako se izostavi prvi broj opsega, znači da seče od prvog karaktera do zadatog, a ako se izostavi drugi broj opsega, *cut* će seći do kraja reda (do poslednjeg karaktera).

Nakon upoznavanja sa još nekim mogućnostima shell-a moći će se od ovih filtara praviti stvarno efikasni programi.

1. **Metakarakteri, regularni izrazi, navodnici**

***Metakarakteri***

Za zamenu u imenu fajlova (slično DOS-u) u Unix/Linux-u mogu se koristiti razni metakarakteri, s tim što ovde postoji mnogo više mogućnosti. Značajna razlika je da uvrštavanje metakaraktera vrši ljuska pre pozivanja naredbe i naredbu pokreće sa pravim imenima fajlova.

1. **Zadatak**

Isprobati naredbu *echo \** koja je ekvivalentna komandi *ls*!

[student@drvo1 myfilms]$ **echo \***

[student@drvo1 myfilms]$ **ls**

Npr. ako u aktuelnom direktorijumu postoje fajlovi ’marija’, ’miroslav’ i ’tamara’ sa naredbom ’cat m\*a\*’ shell će naredbu *cat* pozvati sa dva parametra, kao da smo upisali naredbu ‘cat marija miroslav’ (fajl 'tamara' ne valja jer ne počinje karakterom *m*).

Najrasprostraniji metakarakter je **\***, značenje: 0 ili više proizvoljnog karaktera. Može se koristiti bilo gde u imenu fajla npr. "\*a\*b\*", sa značenjem: svako ime fajla u kojima se javlja negde karakter "a" a posle njega negde i "b".

Drugi popularan metakarakter je **?** sa kojim se zamenjuje tačno jedan proizvoljni karakter.

Može se zadati i skup: izrazom "?[rau]?" pokazujemo na fajlove sa 3 karaktera u kojima je drugi karakter *a* ili *r* ili *u* a prvi i treći karakteri su proizvoljni (npr. kad, sad, trn, kum, ...). [0-9A-Z] – ovaj skup predstavlja jedan karakter koji je ili broj ili veliko slovo.

[^0-9A-Z] – karakter **^** iza otvorene uglaste zagrade označava komplement zadatog skupa (negacija): shell će uzeti u obzir sve karaktere koji ne pripadaju datom skupu.

1. **Zadatak**

U direktorijumu *myfilms* izlistati sve filmove gde je negde u naslovu filma slovo *a* zatim negde slovo *f*.

[student@drvo1 myfilms]$ **ls \*a\*f\***

009 - Monty Python's The Meaning of Life (1983).htm

015 - Austin Powers International Man of Mystery (1997).htm

026 - Shaft (2000).htm

...

1. **Zadatak**

U direktorijumu *myfilms* izlistati sve filmove gde je u rednom boju filma poslednji (treći) broj 2.

[student@drvo1 myfilms]$ **ls 0?2\***

002 - Reservoir Dogs (1992).htm

012 - Clockwork Orange, A (1971).htm

022 - Fabuleux destin d'Amlie Poulain, Le (2001).htm

...

1. **Zadatak**

U direktorijumu *myfilms* izlistati sve filmove od rednog broja 10 do 39.

[student@drvo1 myfilms]$ **ls 0[123]\***

010 - Vanilla Sky (2001).htm

011 - Family Man, The (2000).htm

...

039 - Dead Man (1995).htm

1. **Zadatak**

U direktorijumu *myfilms* izlistati filmove sa rednim brojevima 013-057; rezultat upisati u fajl *myfilms-part.txt*!

[student@drvo1 myfilms]$ **ls 01[3-9]\* 0[2-4]\* 05[0-7]\* > myfilms-part.txt**

[student@drvo1 myfilms]$ **cat myfilms-part.txt**

013 - Natural Born Killers (1994).htm

014 - Contact (1997).htm

...

057 - Mulholland Dr\_ (2001).htm

***Regularni izrazi***

Dizajneri sistema Unix/Linux su napravili regularne izraze (regular expressions) da bi programi kao *grep* mogli pretraživati po "zadatim uslovima".

Regularni izrazi imaju mnogo sličnosti sa metakarakterima u imenima fajlova. Regularni izraz je specijalni niz karaktera koji naredba *grep* i mnoge druge naredbe tumače kao uzorak. Ako data reč odgovara uzorku kaže se da odgovara regularnom izrazu.

Najjednostavniji slučaj je kada regularni izraz ne sadrži nijedan specijalni karakter. Ovakav izraz odgovara samom sebi.

[student@drvo1 myfilms]$ **ls | grep oper**

008 - Starship Troopers (1997).htm

Ova naredba pronalazi samo film sa rednim brojem 008. Sledeća naredba ispisuje filmove gde negde postoji slovo *o* ili *m* zatim sledi niz karaktera *per*:

[student@drvo1 myfilms]$ **ls |grep [om]per**

008 - Starship Troopers (1997).htm

041 - Romper Stomper (1992).htm

Specijalni karakteri koji mogu da se javljaju u regularnim izrazima:

**c**

Svaki karakter odgovara samom sebi.

**\c**

Neutrališe specijalno značenje karaktera *c*. Koristi se kada se baš neki specijalni karakter traži.

**^**

Uzorak je na početku reda. Takvi redovi teksta odgovaraju ovom regularnom izrazu koji počinju sa reči koja je iza znaka ^.

**$**

Kao prethodni karakter, ali se u ovom slučaju uzorak postavlja na kraj reda.

**.**

Svaki karakter odgovara tački sem karaktera za novi red.

**[...]**

Svaki karakter u uglastim zagradama odgovara ovom regularnom izrazu (znači zamenjuje jedan karakter iz niza karaktera u zagradi).

**[^...]**

Ovom regularnom izrazu odgovara svaki karakter sem onih iz uglaste zagrade.

**[n-n]**

Iz date oblasti svaki karakter odgovara regularnom izrazu.

**\***

Karakter pred zvezdicom može da se pojavljuje u reči u bilo kom broju (i nula puta). Npr. a\*: mama, maama, maaaaa, a...

1. **Zadatak**

Iz fajla *sesir* ispisati svaki red koji počinje znakom ^!

Regularni izrazi se moraju staviti pod navodnike.

[student@drvo1 myfilms]$ **grep '^\^' sesir**

Prvi znak ^ označava da izraz posle njega mora da bude na početku reda a drugi znak je sam karakter koji se traži i čije se specijalno značenje ukida pomoću znaka \.

1. **Zadatak**

Izlistati takve direktorijume u folderu */home* u koje svako može ući! Koristiti filtar *grep* i regularne izraze!

[student@drvo1 myfilms]$ **ls -l /home/ |grep '^d........x'**

drwxr-xr-x 2 root root 12288 Apr 28 2008 myfilms

drwxrwxrwx 4 student student 4096 Apr 27 08:13 student

Ovo je komplikovan ali praktičan primer. Iz izlaza naredbe *ls -l* ispisuju se na ekran samo oni redovi koji zadovoljavaju sledeće uslove: počinju karakterom *d*, od dru­gog do devetog mesta može biti bilo koji karakter a na desetom mestu je karakter *x*.

Na ovaj način se dobija lista onih poddirektorijuma u koje svako može ući.

1. **Zadatak**

Iz izlaza naredbe *ls* u direktorijumu *myfilms* izvaditi one redove koji se ne završavaju karakterom *m*!

[student@drvo1 myfilms]$ **ls -l | grep '[^m]$'**

total 7596

-rw-r--r-- 1 student student 39 Apr 27 08:30 imaproblema

-rw-r--r-- 1 student student 1575 Apr 28 12:38 myfilms-part.txt

-rw-r--r-- 1 student student 6758 Apr 27 08:18 myfilms.txt

***Navodnici***

Specijalne karaktere regularnih izraza i shell tumači po svojim pravilima. Ovo je generalno jako korisno ali ne i u slučaju regularnih izraza, zato se regularni izraz mora "braniti" od shell-a. Ako se regularni izraz stavlja između apostrofa ('), shell će ga predati naredbi *grep* u nepromenjenom obliku.

Ako se koriste navodnici ("), shell pregleda regularni izraz, zamenjuje sve pojave promenljivih sa njihovim vrednostima i tako predaje regularni izraz naredbi *grep*.

Treći navodnik koji se može koristiti je `. Regularne izraze koji su pod takvim navodnicima shell pokušava da izvrši kao naredbe i rezultat izvršavanja predaje naredbi *grep*.

1. **Zadatak**

Napraviti tekstualni fajl i analizirati na njemu uticaj navodnika!

[student@drvo1 myfilms]$ **cat >animals**

$mis je jedna mala zivotinja.

microsoft mouse

pas je velik u odnosu na mrava

Da se isprobaju promenljive shell-a napraviti promenljivu *mis*:

[student@drvo1 myfilms]$ **mis=mouse**

[student@drvo1 myfilms]$ **grep '$mis' animals**

$mis je jedna mala zivotinja.

[student@drvo1 myfilms]$ **grep "$mis" animals**

microsoft mouse

Dok je naredba *grep* kao ulaz dobila regularni izraz *$mis* u slučaju apostrofa, u slučaju navodnika je dobila vrednost shell promenljive odnosno reč *mouse* – zato se pojavio red "microsoft mouse" nakon izvršavanja druge naredbe *grep*.

Unix/Linux koristi i treći navodnik ` (na najvećem broju tastatura nalazi se u gornjem levom uglu). Ispisati reč pas pomoću naredbe *echo* i to predati naredbi *grep*!

[student@drvo1 myfilms]$ **grep `echo pas` animals**

pas je velik u odnosu na mrava

1. **Zadatak**

Ispisati na ekran broj fajlova u aktivnom direktorijumu (koristiti kosi navodnik)!

[student@drvo1 myfilms]$ **echo U ovom direktorijumu ima `ls | wc | cut –c4-12` fajlova.**

U ovom direktorijumu ima 111 fajlova.

Šta je sadržaj gornje naredbe? Za podsećanje: opcija *-c* naredbe *cut* seče iz reda karaktere na datoj poziciji.

**Neutralisanje specijalnih karaktera**

Ponekad postoji potreba za tim da se neki karakteri sačuvaju od uticaja shell-a koji bi ih zamenio: značenje svakog karaktera se može neutralisati pomoću karaktera \ (backslash). Ako se želi pisati u fajl čije ime sadrži karakter \*, to se može uraditi na sledeći način: "cat >ab\\*c". Duži tekst se "čuva" pomoću apostrofa '. Ovo može biti korisno ako se naredbom *grep* traži izraz koji sadrži razmak. Kad bi napisali tekst koji sadrži razmak u naredbi, shell bi to protumačio tako kao da je zadato više argumenata. Naredba " ls '? \*' " izlistava (malo čudan) fajl pod imenom "znak pitanja razmak zvezdica" (ako uopšte postoji tako nešto u aktuelnom direktorijumu), a ne traži prvo imena fajlova koji se sastoje od samo jednog karaktera i zatim sva imena fajlova. Sličnu ulogu igraju navodnici " sa razlikom da se ne čuvaju od zamene promenljive okruženja.

1. **Zadatak**

U početnom direktorijumu kreirati fajl pod imenom \*!

[student@drvo1 myfilms]**cat > ~/\\***

zvezde na nebu...

Pomoću karaktera backslash je neutralisano značenje sprecijalnog karaktera \*, a znak ~ zamenjuje putanju početnog direktorijuma.

1. **Zadatak**

Proveriti da li postoji fajl pod imenom \* u početnom folderu!

[student@drvo1 myfilms]$ **ls "${HOME}/\*"**

/home/student/\*

Navodnici su štitili od specijalnog značenja zvezdicu ali su omogućili zamenu vrednosti promenljive okruženja HOME.