Priprema za studentsku praksu

Zadatci – Linux Administrator

v2.0

# Teorijski dio

Prođite kroz listu pojmova iz prvog stupca tablice ispod s ciljem da upoznate, proučite i razumijete što je moguće više pojmova.

|  |  |
| --- | --- |
| **UNIX** | |
| **Uvod** | |
| Konzola | Konzola ili Terminal je program koji otvara prozor i omogućava interakciju sa shell-om. Postoje razni konzolni emulatori poput: gnome-terminal, konsole, xterm, rxvt, eterm, terminator i drugi. |
| Shell | Shell je korisničko CLI sučelje za interakciju s kernelom tj. interaktivni alat za pokretanje programa, obradu teksta, manipulaciju datotekama i datotečnim sustavima, upravljanje procesima, itd. U većini distibucija se koristi bash (Bourne Again Shell). |
| Varijabla | Varijabla je skladište u kojemu se nalazi podatak. Ima svoje ime, tip varijable i vrijednost koja se može mijenjati tijekom rada s varijablom. |
| Apsolutna putanja | Apsolutna (globalna) putanja počinje od korijenskog direktorija prolazeći pritom kroz sve podmape i završava imenom dokumenta na koji se putanja odnosi. |
| Relativna putanja | Kod relativne putanje, početna točka nije korijenski direktorij, već direktorij u kojem se nalazi dokument. |
| Vlasnička struktura direktorija i datoteka | Postoji hijerarhijska struktura datotečnog sustava, početak datotečnog sustava je root (korijen) i označava se s '/'. Ispod root direktorija nalaze se direktoriji koji sadrže poddirektorije i datoteke. |
| Prava pristupa direktorijima i datotekama | "Super User" korisnički račun administrira ostale korisničke račune. Svaki korisnik ima uid i gid. Postoje tri razine dodjeljivanja prava pristupa: user, group i other (ugo). Vrste prava pristupa su read, write i execute (r-w-x). |
| Special characters | Special characters su znakovi koji imaju posebno značenje shell-u te ih se drugačije interpretira. Primjer je '#' koji označava početak komentara, neki od ostalih special char-ova su '&', ';', '|', '?', '$', '~'. |
| Wildcards | Wildcard je znak koji se koristi kao zamjena za neku klasu znakova u pretraživanju, povećavajući fleksibilnost i efikasnost pretraživanja. Najčešće korištena wildcard je asterisk (\*), postoje još '?' wildcard te par uglatih zagrada '[]'. |
| Korisnički profil | Korisnički profil se sastoji od imena i zaporke, ima pristup vlastitom (home) direktoriju te ima ograničen pristup. Podaci računa se nalaze u datoteci /etc/passwd i sadrže: ime korisnika, oznaka polja za zaporku, UID, GID, korisnički podaci, putanja home direktorija i oznaka ljuske. |
| „Cron jobs“ | Cron omugućuje korisnicima pokretanje naredbi i skripti u točno određeno vrijeme. Moguće je zakazati skripte da se izvršavaju periodički. |
| **Specijalni direktoriji** | |
| / | Predstavlja root direktorij. Forward slash se koristi i kao filename path separator. |
| . | Odnosi se na trenutni direktorij. |
| .. | Odnosi se na roditeljski direktorij trenutnog direktorija. |
| ~ | Odnosi se na home direktorij korisnika. |
| **Naredbe za manipulaciju direktorijima i datotekama** | |
| Izlistanje sadržaja direktorija | Naredbom **ls** se ispisuje sadržaj direktorija. Ako se ne navede argument, podrazumijeva se da se radi o radnom direktoriju. |
| Kreiranje direktorija | Direktorij se kreira naredbom **mkdir**, uz opciju **-p** stvorit će se svi direktoriji navedeni kao dio puta koji je naveden kao argument. |
| Kretanje kroz direktorije | Naredbom **cd** mijenjamo radni direktorij, kao argument navodimo ime novog radnog direktorija. |
| Kopiranje direktorija i datoteka | Naredba **cp** služi za kopiranje datoteka i direktorija, treba navesti dva argumenta: izvor i destinaciju. |
| Preimenovanje direktorija i datoteka | Naredba **mv** služi za premještanje ili preimenovanje direktorija i datoteka. Potrebno je navesti izvor i destinaciju. |
| Brisanje direktorija i datoteka | Naredbe **rmdir** i **rm** služe za brisanje direktorija i datoteka. Opcijom **-r** rekurzivno se uklanjaju datoteke i direktoriji te se mogu obrisati i direktoriji koji nisu prazni. |
| Pretraživanje po nazivu direktorija i datoteka | Pretraživanje se vrši naredbom **find**, navodi se direktorij od kojeg počinje pretraživanje i izraz kojim se određuje koje se datoteke traže. Obično je taj izraz **-name**, pretraživanje po imenu datoteke. |
| Ispisivanje sadržaja datoteke | Naredbe **cat**, **more** i **less** ispisuju sadržaj datoteka. **Cat** redom ispisuje sadržaj datoteke koja je navedena kao argument. Naredbe **more** i **less** ispisuju sadržaj datoteka koje su navedene kao argumenti ekran po ekran. |
| Pretraživanje sadržaja datoteke | Naredbom **grep** pretražujemo sadržaj datoteka za jedan ili više regular expression-a. Naredbi se predaje traženi string i ime datoteke. Moguće je predati više stringova i više datoteka. |
| Usporedba sadržaja datoteka | Naredbom **diff** se uspoređuje sadržaj datoteka, ispisuje se njihova razlika za svaku liniju. Naredbi se predaju imena prve i druge datoteke. |
| Promjena prava pristupa direktoriju / datoteci | Prava pristupa postavljamo naredbom **chmod**, postoje dva načina: pomoću simboličkih oznaka i pomoću numeričkih oznaka. Simboličke oznake obuhvaćaju tri razine pristupa i tri vrste prava pristupa, a vrsta izmjene se zadaje operatorima **+**, **-** i **=**. Kod numeričkih oznaka svakoj vrsti pristupa dodijeljena je jedna numerička vrijednost. |
| Promjena vlasništva nad direktorijem / datotekom | Naredbom **chown** se mijenja vlasnik direktorija ili datoteke. Potrebno je navesti novog vlasnika dokumenta i dokument. |
| **Ostale naredbe** | |
| „Pomoć“ za neku naredbu | Pomoć za određenu naredbu se dobiva na način da se nakon imena naredbe doda **-h** za kratki opis i sintaksu naredbe. Za detaljniju pomoć, dodaje se –-**h**. Postoji i naredba **man** kojoj se kao argument predaje naredba, ona ispisuje detaljna uputstva za predanu naredbu. |
| Ispis teksta na konzolu | Naredbom **printf** ispisuje se formatirani tekst na konzolu. Naredbom **cat** ispisuje se sadržaj datoteke na konzolu. |
| Zauzeće diska | Zauzeće diska se ispisuje naredbom **df.** |
| Pregled stanja sustava (procesor i memorija) | Naredbom **top** moguće je provjeriti memorijsku i cpu zauzetost po pojedinom procesu. |
| Izlistanje povijesti naredbi | Tipkama za gore/dolje je moguće scroll-ati kroz prethodno korištene naredbe. Naredbom **history** se ispisuje bash log svih korištenih naredbi. |
| Izlistanje svih procesa | Naredba **ps** izlistava sve pokrenute procese. Daje informacije o PID, TTY, TIME i CMD svakog procesa. |
| Izlistanje suspendiranih i pozadinskih procesa | Naredbi **ps** se može dodati **-aux** flag, tada će naredba izlistati suspendirane i pozadinske procese. |
| Prekidanje procesa | Proces se može prekinuti naredbom **kill**. Naredbi je potrebno predati id procesa (PID) i flag. Ukoliko se ne preda flag, podrazumijeva se standardni **-15** flag. Ponekad takav flag nije dovoljan te se koristi **-9** za prekidanje procesa. |
| Preusmjeravanje „outputa“ | Ukoliko želimo preusmjeriti output sa standardnog outputa koristimo operator **>** te recimo ime datoteke u koju želimo preusmjeriti output. Ukoliko ne želimo zamijeniti sadržaj datoteke novim outputom, već samo dodati sadržaj, koristimo operator **>>**. |
| Preusmjeravanje „inputa“ | Kao što se može preusmjeriti output programa u datoteku, također se može preusmjeriti input naredbe iz datoteke. Koriste se operatori **<** i **<<**. |
| Povezivanje naredbi („pipe“) | Pipes služe za preusmjeravanje streama jednog programa drugom. Standardni output jednog programa se šalje drugom programu, podaci prvog programa se ne prikazuju na terminalu. Pokazuju se samo filtrirani podaci drugog programa. |
| Prečac „Ctrl + C“ | Prečac Ctrl+C se koristi za prekidanje procesa signalom SIGINT, stoga ga program može prekinuti. |
| Naredba „tail“ | Naredba **tail** ispisuje zadnji dio datoteke. Standardno ispisuje zadnjih 10 redova. Naredbi se predaju flag i ime datoteke. |
| Naredba „sed“ | Naredba **sed** se koristi za modifikaciju pojedinog reda datoteke ili streama, zamjenjujući odabrane dijelove reda. Također se koristi za filtriranje jer je moguće izdvojiti redove koji sadrže traženi string. |
| Naredba „awk“ | Naredba **awk** služi za procesiranje ili analiziranje tekstualnih datoteka. Predaje se obrazac i akcija koja će se obaviti na predanoj input datoteci te output datoteka u koju će se spremiti rezultat. |
| Naredba „tar“ | Naredba **tar** se koristi za stvaranje, održavanje, modificiranje i ekstraktiranje datoteka arhiviranih u tar formatu. Prvi argument je funkcija, zatim slijedi ime arhiva i datoteke koje se arhiviraju. |
| Naredba „mount“ | Naredba **mount** povezuje datotečni direktorij uređaja ili particije na datotečno stablo Linux sustava. Naredba povezuje na mount point unutar /mnt direktorija. |
| Naredba „umount“ | Obrnuta naredba od prethodne, prekida vezu uređaja ili particije sa datotečnim stablom Linux sustava. |
| Naredba „set“ | Postavlja vrijednosti shell opcija i pozicionih parametara ili prikazuje imena i vrijednosti shell varijabli. |
| Naredba „unset“ | Služi za brisanje varijabli tokom izvršavanja programa. Može ukloniti funkcije i shell varijable. |
| Naredba „export“ | Naredba **export** postavlja lokalne shell varijable u globalne varijable. |
| Naredba „alias“ | Naredba **alias** omogučava izvršavanje svake naredbe ili grupe naredbi unošenjem predefiniranog stringa. Kratak string može zamjeniti veoma dugu naredbu, što olakšava razumljivost koda. |
| Naredba „cut“ | Naredba **cut** uklanja dijelove svake linije iz datoteke ili datoteka. |
| Regular expressions | Regular expressions služe za uspoređivanje stringa s određenim obrascem u svrhu utvrđivanja valjanosti stringa. Za usporedbu se koristi operator "=~". Vraća vrijednost 0 pri poklapanju stringa i regexa, u suprotnom vraća 1. Može se koristiti za npr. provjeru unesene email adrese. |
| **Osnovni preglednici** | |
| Vrste osnovnih preglednika | U osnove preglednike spadaju naredbe poput **head** i **tail**, koje ispisuju prvih ili zadnjih 10 redaka. Također spadaju naredbe **more** i **less**. |
| Naredbe u „less“ pregledniku | Tipkama g i G se pomiče preglednik na specificiranu liniju. Moguće je postaviti marker tipkom m. Znakom / se traži string ili regular expression. |
| **vi editor** | |
| Uređivanje teksta | Vi editor posjeduje dva načina rada: command način i insert način. Za uređivanje testa potrebno je ući u insert način pritiskom na tipku i, a ili o. |
| Snimanje promjena | Naredbom **:w** spremamo promjene u istu datoteku, nakon **:w** moguće je upisati ime nove datoteke u koju će se spremiti promjene. |
| Izlazak iz editora | Izlazi se naredbama :wq za spremanje promjena u istu datoteku ili s **:q** za izlazak iz editora. |
| **SHELL SKRIPTE** | |
| **Uvod** | |
| Argumenti | Svi navedeni argumenti nakon imena skripte se prosljeđuju skripti kao niz varijabli. Unutar skripte parametri se mogu dohvatiti na nekoliko načina, recimo, **$@** vraća niz stringova parametara. **$1**, **$2**...**$n** vraća n-ti argument predan skripti, dok **$0** vraća ime same skripte. |
| Funkcije | Shell funkcije mogu vraćati vrijednost i/ili stvoriti output. Funkcija može vratiti vrijednost na 4 načina: promjena stanja varijabli, naredbom **exit** s kojim završava skriptu, naredbom **return** s kojom završava funkciju ili ispisivanjem outputa na stdout. |
| Environment varijable | Svaka pokrenuta skripta nasljeđuje environment varijable dostupne roditeljskom shellu. |
| Specijalne varijable | Specijalne varijable su rezervirane za specifičnu funkciju. Neke od specijalnih varijabli su **$$** (PID trenutnog shella), **$0** (ime trenutne skripte), **$?** (exit status zadnje izvršene naredbe), itd. |
| Komentari | Komentari započinju znakom **#**, nakon znaka svi znakovi do kraja reda spadaju u komentar. Moguće je započeti komentar na početku reda ili poslije određene naredbe. |
| **Osnovni operatori** | |
| Aritmetički operatori | U aritmetičke operatore spadaju +, -, /, \*, %, ++, --, \*\*. Aritmetička evaluacija se izvršava u formatu $((expression)). |
| Relacijski operatori | Relacijski operatori uspoređuju dvije vrijednosti, postoje: -**eq**, **-ne**, **-gt**, **-lt**, **-ge**, **-le**. Svi relacijski izrazi se nalaze unutar uglatih zagrada koje su odvojene zarezom od izraza. |
| Boolean operatori | U boolean operatore spadaju **!** (logička negacija), **&&** (logičko i) i **||** (logičko ili). Operator **&&** povezuje naredbe na način da se sljedeća naredba izvršava ukoliko je prethodna uspješno izvršena. Operator **||** dopušta izvođenje sljedeće naredbe ukoliko prethodna nije izvršena. |
| String operatori | Postoje **${#string}** koji vraća duljinu stringa, **${string:position:length}** izdvaja duljinu stringa počevši od neke pozicije. Operator **${string/pattern/replacement}** pronalazi prvi obrazac te ga zamjenjuje drugim obrascem. |
| Operatori datoteka | Neki od operatora su **-b** (block special datoteka?), **-c** (character special datoteka), **-d** (direktorij?), **-r** (pristup čitanju?), itd. |
| **Kontrola toka: odlučivanje** | |
| if – fi | If u bashu ima sintaksu koja zahtjeva format odlučivanja u kojem se navodi ključna riječ **if** te se u uglastim zagradama navodi uvjet. Slijede ključne riječi **then** i **fi**, između kojih se nalazi blok koji se izvodi u slučaju da je uvjet ispunjen. |
| if – else – fi | U if-else odlučivanju se izvodi drugi blok ako uvjet nije zadovoljen. Između ključnih riječi **then** i **fi**, nalazi se ključna riječ **else**, nakon koje slijedi drugi blok naredbi. |
| if – elif – else – fi | U slučaju da postoji više uvjeta koji vode drugačijim putevima, koristi se if-elif-else odlučivanje. Nakon ključne riječi **elif**, navodi se uvjet i ključna riječ **then** te blok naredbi koji se izvršava za taj slučaj. |
| case – esac | Nekada je potrebno odabrati različite puteve u ovisnosti je li varijabla odgovara nekom od obrazaca. Sintaksa nalaže ključnu riječ **case**, ime varijable i ključnu riječ **in**. Nakon toga se navode obrazci sa svojim blokovima naredbi. |
| **Kontrola toka: petlje** | |
| for do – done | For petlja se u bashu izvodi pomoču ključnih riječi **for**, **in**, **do** i **done**. Nakon riječi **for** slijedi ime varijable, zatim riječ **in** te red vrijedosti kroz koju varijabla prolazi. Između riječi **do** i **done** se nalazi blok koji se izvršava u svakoj iteraciji petlje. |
| while do – done | Sliča prethodnoj petlji, while petlja izvršava blok sve dok je uvjet ispunjen, moguče je da se petlja uopće ne izvrši. Nakon riječi **while** slijedi uvjet u uglatim zagradama, a između **do** i **done** se nalazi blok naredbi. |
| until do – done | Until petlja nastavlja izvoditi blok naredbi dokle god je uvjet istinit, jednom kad uvjet postane false, petlja završava. |
| select do – done | Select petlji se predaje slijed kroz koji se vrši iteracija te se prikazuje kao numerirani izbornik. Odabirom nekog broja, varijabli se pridodaje vrijednost odabranog izbora. |
| **Kontrola toka: bezuvjetni skok** | |
| break | **break** služi za prekidanje petlje prije normalnog završetka petlje. Koristi se kad nije poznato koliko puta se petlja treba izvršiti. |
| continue | **continue** se koristi kako bi petlja nastavila s sljedećom iteracijom. Za razliku od **break** izjave, **continue** ne završava s petljom, već samo nastavlja s sljedećom iteracijom. |
| return | Izjava **return** vraća vrijednost funkcije. **return** vraća exit status čija vrijednost mora biti broj. |
| **SQL** | |
| **Općenito** | |
| table | Temelj svakog DBMS-a je objekt baze zvan table. Svaka baza podataka sastoji se od jedne ili više tablica koje sadrže podatke baze. Svaka tablica ima svoje ime te se sastoji od stupaca i redova. |
| column | Tablica baze sadrži stupce (columns) i retke (rows). Stupci predstavljaju atribute baze poput imena, prezimena, email adrese, itd. |
| view | U SQL jeziku, view je virtualna tablica bazirana na result-setu SQL naredbe. Polja viewa su polja iz jedne ili više tablica baze podataka. |
| index |  |
| constraint |  |
| stored procedure |  |
| trigger |  |
| create | Naredba **create** stvara objekte podataka, uključujući nove baze i tablice, views i stored procedures. |
| drop | Suprotna naredba od **create**, **drop** naredba uklanja objekte podataka, poput baze podataka, tablice, viewa, indeksa ili procedure. |
| select | **Select** naredba se koristi za dohvačanje podataka iz baze, dohvaćeni podaci se spremaju u tablicu rezultata zavni result-set. Sintaksa zahtjeva navođenje ključne riječi **SELECT**, zatim tražene podatke, ključnu riječ **FROM** te naziv tablice. |
| insert | Naredba **insert** unosi nove podatke unutar tablice baze podataka. Navode se ključne riječi **INSERT** **INTO** s nazivom tablice i stupcima, zatim slijedi Ključna riječ **VALUES** s vrijednostima za svaki stupac. |
| update | **Update** naredba se koristi za modificiranje postoječih podataka unutar tablice baze podataka. Moguće je navesti ključnu riječ **WHERE** kako bi se odabralo koji redovi će se modificirati, u protivnome, svi se redovi modificiraju. |
| delete | Naredba **delete** briše postoječe podatke unutar tablice baze podataka. Koriste se ključne riječi **DELETE** **FROM** zajedno s imenom tablice. Ako se ne navede **WHERE** zajedno s uvjetom, svi redovi tablice se brišu. |
| commit | Naredba **commit** stvara trajne promjene na bazi podataka napravljene tijekom trenutne transakcije. Svaki **commit** stvara transakcijsku datoteku te se suprotnom naredbom (**rollback**) može vratiti baza u prijašnje stanje. |
| **Vrste relacija između tablica** | |
| 1 – 1 |  |
| 1 – više |  |
| više - više |  |
| **SVN** | |
| **Općenito** | |
| Check-out |  |
| Commit |  |
| trunk |  |
| branch |  |

# Praktični dio

Riješite navedene zadatke vodeći računa o točnosti rješenja, čitljivosti i razumljivosti koda. Sve skripte koje budete kreirali tijekom priprema trebaju biti nazvane ovako:

zadatak<broj zadatka>\_<proizvoljan naziv>

gdje <broj zadatka> predstavlja dvoznamenkastu brojčanu oznaku zadatka (npr. 03 ili 13), a <proizvoljan naziv> predstavlja vaš naziv te skripte.

Primjer: naziv shell skripte koja je rješenje trećeg zadatka:

zadatak03\_HelloWorld.sh

## Zadatci

1. Otvori neku veliku tekstualnu datoteku pomoću less naredbe:
   * Pomoću prečaca odi na kraj datoteke {g for start, G for EOF}
   * Pretraži neki pojam od kraja prema početku ?{?search}
   * Skoči na početak datoteke {g for start, G for EOF}
   * Pretraži neki pojam od početka prema kraju {/Linux}
2. Kreiraj novu datoteku naziva zadatak02\_MyFirstFile i uredi ju pomoću vi editora:
   * Upiši desetak redova {i za insert mode}
   * Umetni novi red u prvoj liniji datoteke {o za novi red}
   * Obriši zadnju liniju datoteke {dd za brisanje cijele linije}
   * Pretraži neki pojam od početka prema kraju {/string}
   * Izmjeni riječ u sredini datoteke {:5 za postavljanje kursora u 5. red, cw za promjenu rijeći}
   * Spremi sve izmjene i izađi iz editora {:wq za spremanje i izlazak iz editora}
3. Kreiraj tzv. "Hello World" skriptu tj. skriptu koja će nakon pokretanja na konzolu ispisati riječi "Hello World".
4. Kreiraj program koji prima jedan argument – cjelobrojnu vrijednost. Program treba provjeriti je li argument paran ili neparan broj te ispisati poruku na konzolu u obliku „X je paran broj“ ili „X je neparan broj“.
5. Kreiraj program koji za argumente prima početak i kraj nekog intervala. Neka početak intervala bude manji od 10, a kraj intervala veći od 100. Program treba provjeriti jesu li početak i kraj intervala ispravno definirani te ispisati poruku o pogrešci ukoliko nisu. Ukoliko jesu program treba za svaki broj u zadanom intervalu napraviti sljedeće:
   * ukoliko je broj manji ili jednak 50, treba prekinuti procesuiranje tog broja i prijeći na sljedeći
   * ukoliko je broj veći od 50, treba ispisati taj broj na konzolu u obliku „Obradujem X“
   * ukoliko je broj jednak 75, treba prekinuti procesuiranje intervala te ispisati poruku „Obrada gotova“
6. Kreiraj program koji će ispisati koliko ima brojeva djeljivih sa 7 u intervalu između neka dva prirodna broja učitana iz konfiguracijske datoteke. Putanja do konfiguracijske datoteke treba biti argument skripte.
7. Kreiraj program koji će unutar nekog direktorija rekurzivno pronaći sve datoteke čiji naziv počinje s „my\_“ i izmijeniti njihova pristupna prava. Putanja direktorija za pretraživanje i pristupna prava su argumenti skripte.
8. Kreiraj program koji će 10 puta tražiti korisnika unos teksta. Nakon svakog unosa, program treba uneseni tekst upisati u izlaznu datoteku. Nakon završetka programa, izlazna datoteka treba sadržavati svih 10 unosa i samo tih 10 unosa. Putanja izlazne datoteke je argument skripte.
9. Kreiraj program koji treba u izlaznu datoteku ispisati apsolutne putanje svih datoteka s ekstenzijom "txt" unutar nekog direktorija u kojima se nalazi neka tekstualna fraza. Putanja izlazne datoteke i direktorija za pretraživanje te tekstualna fraza su argumenti skripte.
10. Kreiraj program koji će ispisati sve PID-ove procesa pokrenutih od strane nekog korisnika. Ime korisnika je argument skripte.
11. Kreiraj program koji će unutar nekog direktorija rekurzivno pronaći sve datoteke čiji naziv počinje s „my\_“ i obrisati iz njih linije koje počinju s velikim slovom.
12. Kreiraj program koji će primati dva argumenta, od kojih je drugi opcionalan te imati tri funkcije:
    * Funkcija koja provjerava ispravnost broja argumenata (u slučaju krivog broja, poziva prvo funkciju za ispis uputa, a zatim i funkciju za ispis poruke o pogrešci)
    * Funkcija koja ispisuje upute kako ispravno koristiti program
    * Funkcija koja ispisuje poruku o pogrešci i izlazi van iz programa
13. Kreiraj datoteku koja će biti u /etc/cron.d kako bi se periodički pokretale sljedeće skripte:
    * skripta /tmp/script1.sh svaku minutu
    * skripta /tmp/script2.sh svakih 15 minuta
    * skripta /tmp/script3.sh svakih sat vremena u puni sat i 15 minuta
    * skripta /tmp/script4.sh svakih 2 sata
    * skripta /tmp/script5.sh svaki dan u 02:30
14. Kreiraj program koji će koristeći regular expression provjeriti ispravnost naziva datoteka u nekom direktoriju. Putanja direktorija za pretraživanje je argument skripte, a naziv datoteke je ispravan ukoliko su svi navedeni uvjeti zadovoljeni:
    * Prva četiri znaka su „ABCD“ (točno tim redom)
    * Sljedeća četiri znaka su bilo koji brojevi
    * Zatim slijedi podvlaka
    * Sljedeća tri znaka trebaju biti bilo koja velika slova engleske abecede
    * Zatim slijedi crtica
    * Naposljetku slijedi neograničen niz znakova koji uključuju brojeve i bilo koja (velika ili mala) slova engleske abecede

Primjer ispravnog naziva datoteke: ABCD5189\_XPR-SomeText007

1. Napraviti program koji će omogućiti korisniku ispis statistike broja pojedinih riječi unutar datoteke koju navodi kao argument. Riječi trebaju biti poredane po abecedi i treba ignorirati velika/mala slova. Ispod se nalazi primjer ispisa:

U datoteci XX nalaze se sljedece rijeci:

------------------------

Rijec (broj ponavljanja)

------------------------

auto (17)

auta (2)

boje (19)

je (39)

su (110)

------------------------

# Upute za završni izvještaj

Završni izvještaj sadrži rezultat svladavanja teorijske i praktične grupe zadataka:

* Završni izvještaj teorijskog dijela podrazumijeva da u drugi stupac tablice iz poglavlja 1. unesete vaše razumijevanje pojedinog pojma navedenog u prvom stupcu te tablice. Preporuča se pojam opisivati vlastitim riječima na hrvatskom jeziku.
* Za završni izvještaj praktičnog dijela trebate sve skripte spremiti u direktorij PripremaPrakse\_<ImePrezime>. Direktorij sažmite u zip datoteku jednakog naziva (PripremaPrakse\_<ImePrezime>.zip). Naposljetku, tu zip datoteku umetnite na kraj ovog word dokumenta.

Nakon što ste odradili navedena dva koraka spremite ovaj word dokument. Prije nego ga pošaljete provjerite jeste li promijenili ImePrezime iz naziva dokumenta u vaše ime i prezime.