Računalniški praktikum in Osnove računalništva

2. Linux - delo z besedilnimi datotekami

Linux osnovni ukazi - ponovitev

- **man** - za prikaz priročnika
- **clear** - počisti terminal
- **ls** - prikaže vsebino
- **cd** - spremeni direktorij
- **pwd** - pot do direktorija
- **mkdir** - ustvari direktorij
- **rmdir** - odstrani direktorij
- **touch** - ustvari datoteko
- **rm** - odstrani datoteko (direktorij)
- **cp** - kopiraj datoteko
- **mv** - premakni datoteko
- **history** - prikaži zgodovino ukazov

Pravice datotek - ponovitev

V Linuxu so trije tipi lastnikov: uporabnik (ki je ustvaril element), skupina, ostali.

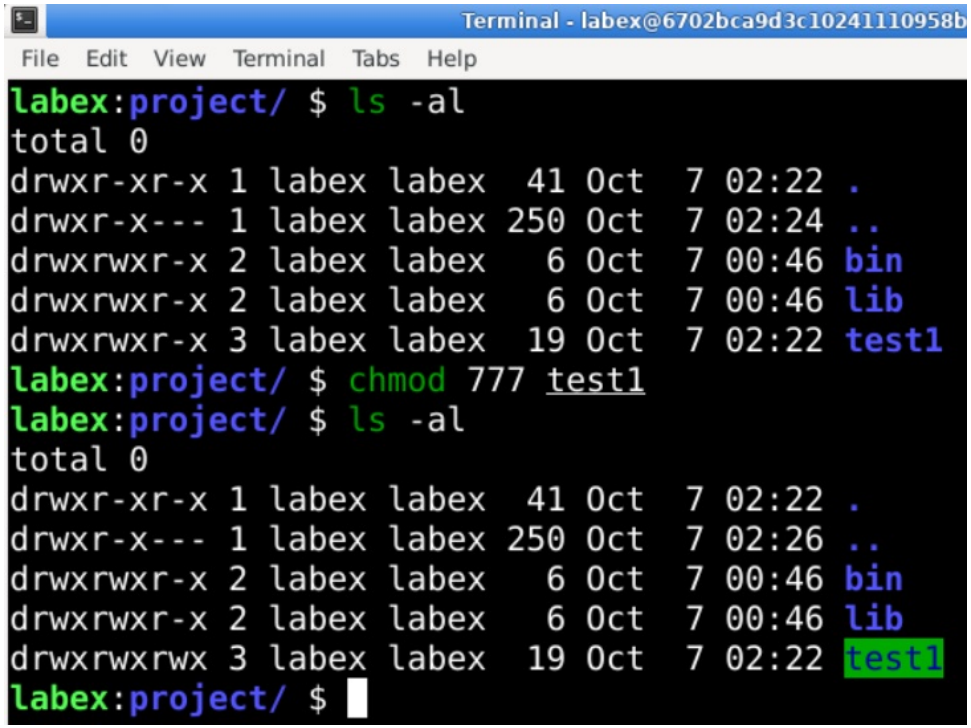
```
drwxr-xr-x 1 labex labex  41 Oct  7 02:22 .
drwxr-x--- 1 labex labex 250 Oct  7 02:24 ..
drwxrwxr-x 2 labex labex   6 Oct  7 00:46 bin
```

Z ukazom ls -l lahko pogledamo pravice (prvih 10 znakov). Prvi znak nam pove tip datoteke (d - direktorij, - navadna datoteka), obstajajo tudi druge oznake. Sledijo po trije znaki za vsak tip uporabnika po vrsti (znaki so po vrsti: r - Read, w - Write, x - eXecute), kjer - pomeni, da uporabnik nima tiste pravice. Pravice lahko podamo tudi v osmiškem zapisu (4 - read, 2 - write, 1 - execute).

Kakšne vrednosti osmiškega zapisa imamo na sliki?

Linux ukazi - chmod - ponovitev

chmod [pravice] [direktorij/datoteka] - nam omogoča spreminjanje pravic direktorijem in datotekam. Pod argument pravice lahko s tremi vrednostmi v osmiškem zapisu podamo pravice.



Delo z besedilnimi datotekami

Ukaz `nano`

- Ukaz **nano** zažene preprost urejevalnik besedila, ki deluje neposredno v terminalu. Uporabimo ga lahko za ustvarjanje novih besedilnih datotek ali urejanje obstoječih.
- Primer: `nano datoteka.txt`
- Če datoteka še ne obstaja, jo urejevalnik ustvari, ko jo shranimo. Če datoteka že obstaja, se odpre njena trenutna vsebina.
- Med urejanjem lahko besedilo vnašamo neposredno s tipkovnico. Na dnu zaslona so prikazane najpogostejše bližnjice za delo z urejevalnikom.

Ukaz `nano`

- Najpomembnejše bližnjice:
  - `Ctrl + O` – shrani spremembe v datoteko (Write Out).
  - `Enter` – potrdi ime datoteke pri shranjevanju.
  - `Ctrl + X` – zapre urejevalnik.
  - `Ctrl + W` – poišče besedilo v datoteki.
  - `Ctrl + K` – izreže trenutno vrstico.
  - `Ctrl + U` – vstavi nazadnje izrezano besedilo.
- Pri izhodu iz urejevalnika lahko program vpraša, ali želimo shraniti spremembe. Če datoteko urejamo prvič, moramo ob shranjevanju potrditi tudi njeno ime.
- Bližnjice v `nano` se običajno zapisujejo z znakom `^`, zato na primer `^X` pomeni `Ctrl + X`.

Ukaz `cat`

- Ukaz **cat** izpiše vsebino ene ali več datotek na standardni izhod, ki je običajno terminal. Najpogosteje ga uporabljamo za hiter pregled kratkih besedilnih datotek.
- Primer: `cat datoteka.txt`
- Ukaz `cat` ne odpre interaktivnega urejevalnika in vsebine datoteke ne spreminja. Če datoteka ne obstaja ali je ne moremo prebrati, izpiše sporočilo o napaki.

Ukaz `cat`

- Ukazu `cat` lahko podamo več datotek. Njihova vsebina se izpiše zaporedno, v istem vrstnem redu, kot so navedene.
- Primer: `cat prva.txt druga.txt`
- Vsebino več datotek lahko združimo tudi v novo datoteko s preusmeritvijo izhoda.
- Primer: `cat prva.txt druga.txt > zdruzeno.txt`

Ukaz `cat`

- Ukaz `cat` omogoča tudi prikaz posebnih znakov in številčenje vrstic.
- Primer: `cat -n beležke.txt`
- Zastavica `-n` oštevilči vse izpisane vrstice.
- Primer: `cat -b beležke.txt`
- Zastavica `-b` oštevilči samo neprazne vrstice.
- Primer: `cat -A beležke.txt`
- Zastavica `-A` prikaže tudi nekatere znake, ki sicer niso neposredno vidni, na primer tabulatorje in oznake konca vrstice. To je uporabno pri preverjanju oblikovanja besedilnih datotek.

Ukaz `more`

- Ukaz **more** omogoča pregledovanje daljših besedilnih datotek po delih. Namesto da bi se celotna vsebina izpisala naenkrat, prikaže toliko besedila, kolikor ga gre na zaslon terminala.
- Primer: `more datoteka.txt`
- Ko zapolnimo trenutni zaslon, se izpis ustavi. Nato lahko nadaljujemo s pregledovanjem vsebine.
- Omejitev: `more` je namenjen predvsem pomikanju naprej. Za bolj prilagodljivo pregledovanje, zlasti kadar se želimo pomikati tudi nazaj ali iskati po vsebini, je primernejši `less`.

Ukaz `more`

- Med pregledovanjem vsebine lahko uporabljamo naslednje tipke:
  - `Space` – pomik naprej za en zaslon.
  - `Enter` – pomik naprej za eno vrstico.
  - `b` – pomik nazaj za en zaslon, če ga uporabljena različica podpira.
  - `q` – zaključek pregledovanja in vrnitev v terminal.
- Opozorilo: Nabor podprtih tipk se lahko nekoliko razlikuje med različicami programa in terminalskimi okolji.

Ukaz `more`

- Ukaz `more` lahko uporabimo tudi za pregledovanje izhoda drugega ukaza. Pri tem uporabimo operator cevovoda (pipeline) `|`, ki izhod prvega ukaza posreduje kot vhod drugemu ukazu.
- Primer: `ls -l | more`
- Ukaz `ls -l` izpiše podroben seznam vsebine trenutnega direktorija, ukaz `more` pa omogoči postopno pregledovanje izpisa.
- Tak način uporabe je posebej koristen, kadar ukaz izpiše več vrstic, kot jih lahko hkrati vidimo v terminalu.

Ukaz `less`

- Ukaz **less** omogoča interaktivno pregledovanje daljših besedilnih datotek. Podobno kot more prikazuje vsebino po delih, vendar omogoča bolj prilagodljivo navigacijo, vključno s pomikanjem naprej in nazaj.
- Primer: `less datoteka.txt`
- Ukaz odpre pregledovalnik vsebine datoteke. Datoteke pri tem ne spreminja, po končanem pregledovanju pa se vrnemo v terminal.
- Prednost ukaza `less`: omogoča učinkovito pregledovanje dolgih izpisov, ne da bi morali celotno vsebino izpisati naenkrat ali jo odpreti v urejevalniku.

Ukaz `less`

- Med pregledovanjem vsebine lahko uporabljamo naslednje tipke:
  - `Space` – pomik naprej za en zaslon.
  - `Enter` ali `j` – pomik naprej za eno vrstico.
  - `b` – pomik nazaj za en zaslon.
  - `k` – pomik nazaj za eno vrstico.
  - `g` – pomik na začetek datoteke.
  - `G` – pomik na konec datoteke.
  - `q` – izhod iz pregledovalnika.
- Opozorilo: Nabor podprtih tipk se lahko nekoliko razlikuje med različicami programa in terminalskimi okolji.
- Tako kot `more` lahko tudi `less` uporabimo za pregledovanje izhoda drugega ukaza.

Ukaz `head`

- Ukaz **head** izpiše začetni del datoteke. Privzeto prikaže prvih deset vrstic, kar je uporabno, kadar želimo hitro preveriti vsebino daljše datoteke, ne da bi izpisali celotno vsebino.
- Primer: `head datoteka.txt`
- Število izpisanih vrstic lahko določimo z zastavico `-n`.
- Primer: `head -n 5 datoteka.txt`
- Podobno lahko pregledamo začetni del več datotek.
- Primer: `head -n 3 prva.txt druga.txt`
- Ukaz izpiše prve tri vrstice vsake datoteke in pri tem označi, kateri datoteki pripada posamezen izpis.

Ukaz `tail`

- Ukaz **tail** izpiše končni del datoteke. Privzeto prikaže zadnjih deset vrstic, kar je uporabno, kadar želimo hitro preveriti vsebino dnevniških datotek, kjer so novejši zapisi na koncu.
- Primer: `tail datoteka.txt`
- Število izpisanih vrstic lahko določimo z zastavico `-n`.
- Primer: `tail -n 5 datoteka.txt`
- Podobno lahko pregledamo končni del več datotek.
- Primer: `tail -n 3 prva.txt druga.txt`
- Ukaz izpiše zadnje tri vrstice vsake datoteke in pri tem označi, kateri datoteki pripada posamezen izpis.

Ukaz `tail`

- Ena najuporabnejših možnosti ukaza `tail` je zastavica `-f`, ki omogoča sprotno spremljanje sprememb datoteke.
- Primer: `tail -f datoteka.txt`
- Ukaz najprej izpiše zadnjih deset vrstic datoteke, nato pa čaka na nove podatke. Ko se na konec datoteke dodajo nove vrstice, jih izpiše tudi v terminalu.
- Tak način uporabe je pogost pri spremljanju dnevniških datotek med izvajanjem programa ali odpravljanjem napak.
- Spremljanje običajno prekinemo s kombinacijo tipk `Ctrl + C`.

Ukaz `wc`

- Ukaz **wc** (word count) omogoča štetje vrstic, besed in bajtov v besedilni datoteki. Uporaben je za hitro pridobivanje osnovnih informacij o obsegu datoteke.
- Primer: `wc datoteka.txt`
- Privzeti izpis vsebuje tri številske vrednosti v zaporedju: število vrstic, število besed in število bajtov.
- Na koncu izpisa je navedeno ime datoteke. Pri tem ukaz šteje vrstice glede na znake za konec vrstice, besede pa glede na zaporedja znakov, ločena s presledki.

Ukaz `wc`

- Z zastavicami lahko določimo, kateri podatek želimo izpisati.
- Zastavica `-l` pomeni, da izpišemo število vrstic.
- Zastavica `-w` pomeni, da izpišemo število besed.
- Zastavica `-c` pomeni, da izpišemo število bajtov.
- Zastavica `-m` pomeni, da izpišemo število znakov.
- Število znakov se lahko razlikuje od števila bajtov, zlasti pri besedilu, ki vsebuje znake, zapisane z več bajti, na primer nekatere črke s strešicami.

Ukaz `sort`

- Ukaz **sort** razvrsti vrstice besedilne datoteke in rezultat izpiše na standardni izhod. Privzeto razvršča vrstice leksikografsko, pri čemer je vrstni red lahko odvisen tudi od jezikovnih nastavitev okolja.
- Primer: `sort datoteka.txt`
- Ukaz `sort` privzeto ne spreminja izvorne datoteke, temveč samo izpiše razvrščen rezultat.
- Za številsko razvrščanje ali pa razvrščanje v obratnem vrstnem redu potrebujemo zastavici `-n` (numeric) in `-r` (reverse).

Ukaz `sort`

- Če vrstice vsebujejo več podatkov, lahko določimo stolpec (ali več stolpcev) po katerem želimo razvrščati. Zastavica `-k` določa ključ razvrščanja. Za vsak stolpec lahko postavimo ločeno zastavico `-k` v vrstnem redu prioritete razvrščanja.
- Razvrščanje po drugem stolpcu, ki vsebuje števila izgleda tako.
- Primer: `sort -k2,2n datoteka.txt`
- Ukaz razvrsti vrstice po številskih vrednostih v drugem stolpcu, od najmanjše do največje. Z zapisom `-k2,2n` določimo, da je ključ razvrščanja samo drugi stolpec in da ga obravnavamo številsko.

Ukaz `uniq`

- Ukaz **uniq** prepozna **zaporedne** enake vrstice v vhodnem besedilu in jih privzeto izpiše samo enkrat. Uporaben je pri obdelavi seznamov, dnevniških zapisov in drugih besedil, v katerih se podatki ponavljajo.
- Primer: `uniq datoteka.txt`
- Če želimo odstraniti vse podvojene vrstice, moramo enake vrstice najprej postaviti skupaj. To lahko dosežemo z ukazom `sort`.
- Zastavica `-c` izpiše število pojavitev vsake skupine zaporednih enakih vrstic.
- Pomembno: `uniq` sam po sebi ne poišče vseh ponovitev v datoteki, temveč le obravnava zaporedne enake vrstice.

Ukaz `cut`

- Ukaz **cut** omogoča izpis izbranih delov vsake vrstice besedilne datoteke. Uporabimo ga lahko za izbiranje posameznih znakov, položajev ali polj, kadar so podatki zapisani v urejeni obliki.
- Zastavica `-c` določa položaje znakov, ki jih želimo izpisati.
- Primer: `cut -c 1-5 datoteka.txt`
- Ukaz iz vsake vrstice datoteke imena.txt izpiše prvih pet znakov.
- Če želimo izpisati samo tretji znak, uporabimo:
- Primer: `cut -c 3 datoteka.txt`

Ukaz `cut`

- Pogosteje ukaz `cut` uporabljamo za izbiranje stolpcev oziroma polj, ločenih z določenim znakom. Zastavica `-d` določa ločilo, zastavica `-f` pa številko polja, ki ga želimo izpisati.
- Izpis prvega polja, kjer so podatki ločeni z vejico:
- Primer: `cut -d ',' -f 1 datoteka.txt`
- Izpis drugega in tretjega polja:
- Primer: `cut -d ',' -f 2,3 datoteka.txt` ali
  `cut -d ',' -f 2-3 datoteka.txt`

Ukaz `paste`

- Ukaz **paste** združuje vsebino več datotek tako, da njihove ustrezne vrstice izpiše v skupni vrstici. Privzeto posamezna polja loči s tabulatorjem.
- Primer: `paste imena.txt tocke.txt`
- Privzeto je ločilo med polji tabulator. Z zastavico `-d` lahko določimo drugo ločilo.
- Primer: `paste -d ',' imena.txt tocke.txt`

Ukaz `grep`

- Ukaz **grep** omogoča iskanje **vrstic**, ki ustrezajo določenemu besedilu ali vzorcu. Izpiše samo tiste vrstice, v katerih najde ujemanje, zato je uporaben za pregledovanje dnevniških datotek, konfiguracij in drugih besedil.
- Primer: `grep "error" sistem.log`
- Iskanje privzeto razlikuje med velikimi in malimi črkami.
- Iščemo lahko tudi več nizov hkrati na enega od naslednjih načinov:
- Primer: `grep -E "niz1|niz2" datoteka.txt`
- Primer: `grep -e "niz1" -e "niz2" datoteka.txt`
- Primer: `egrep "niz1|niz2" datoteka.txt`

Linux operatorja - > in >>

- Operator > preusmeri izhod ukaza v datoteko, namesto da bi se izpisal na zaslon.
- Primer: `ls -l > datoteka.txt`
- Uporabimo ga lahko s poljubnim ukazom, ki zapisuje v terminal.
- Preusmerjanje je uporabno, kadar želimo rezultate ukaza shraniti za poznejši pregled, nadaljnjo obdelavo ali uporabo v drugem programu.
- Če ciljna datoteka še ne obstaja, se ustvari. Če že obstaja, se njena prejšnja vsebina praviloma prepiše.
- Če želimo obstoječo vsebino ohraniti in na konec dodati nov izpis, uporabimo operator >>.

Linux operator - |

- Operator | (**cevovod** oz. **pipeline**) omogoča, da izhod enega ukaza neposredno posredujemo drugemu ukazu. Tako lahko rezultate enega ukaza dodatno obdelamo, ne da bi jih morali najprej shraniti v datoteko.
- Primer: `cat datoteka.txt | sort`
- Vsebino datoteke prebere z ukazom `cat` in posreduje ukazu `sort`, ki vrstice razvrsti po abecedi.
- Osnovno pravilo: ukaz na levi strani operatorja | ustvari izhod, ukaz na desni strani pa ta izhod prejme kot vhodni podatek.
- Zaporedoma lahko uporabimo tudi več pipeline operatorjev.
- Primer: `cat datoteka.txt | sort | uniq`
- Ukaz prebere datoteko, razvrsti njene vrstice in odstrani zaporedne podvojene vrstice.