Pipeline – Gyakorló feladatsor

Egy lehetséges megoldás

A kedvenc zeneszámaink adatait egy `playlist.csv` nevű állományban tároljuk. A fájl egyes soraiban szereplő, pontosvesszővel elválasztott adatok rendre:

- **az előadó neve**
- **a zeneszám címe**
- **a zeneszám műfaja**
- **a zene hossza (másodpercben).**

Oldd meg az alábbi feladatokat csővezeték segítségével!

1. Összesen hány zene adatait tartalmazza a fájl (azaz hány sorból áll)?

A pipeline-os feladatoknál első lépésben hasznos lehet, ha kiíratjuk a feldolgozandó fájlnak a tartalmát. Ezt a gyakorlaton tanult **`cat`** paranccsal tudjuk megtenni.

```
cservz@debian:~$ cat playlist.csv
Rick Astley;Never Gonna Give You Up;pop;213
Imagine Dragons;Thunder;pop;204
Dragonforce;Through the Fire and Flames;metal;445
Boney M.;Rasputin;pop;284
Steppenwolf;Born To Be Wild;rock;216
Powerwolf;Incense and Iron;metal;240
Smash Mouth;All Star;rock;237
Nirvana;Smells Like Teen Spirit;rock;279
Gloryhammer;The Unicorn Invasion of Dundee;metal;265
Powerwolf;We Are The Wild;metal;221
Imagine Dragons;Radioactive;rock;188
Dschinghis Khan;Moskau;pop;275
Dschinghis Khan;Dschinghis Khan;pop;185
Bonnie Tyler;Total Eclipse of the Heart;pop;334
Gopnik McBlyat;Snakes In Tracksuits;hardbass;261
John Farnham;Thunder In Your Heart;rock;234
```

A feladatunk megszámolni, hogy hány darab sorból áll a fájl. Ha valamit számolni kell egy pipeline-os feladatban, akkor jusson eszünkbe a **`wc`** parancs. Ennek a főbb kapcsolói, amiket a gyakorlaton tárgyalunk:

- **`-l`**: a sorok száma a fájlban
- **`-w`**: a szavak száma a fájlban
- **`-c`**: a bájtok száma a fájlban.

Mivel mi a sorok számára vagyunk kíváncsiak, ezért nekünk a **`-l`** kapcsoló fog kelleni. Pipeline segítségével fűzzük össze a **`cat playlist.csv`** és **`wc -l`** parancsokat (hiszen a **`cat`** parancs kimenetén szeretnénk végrehajtani a sorok megszámolását)!

A feladat megoldása tehát:

```
cservz@debian:~$ cat playlist.csv | wc -l
16
```

2. Hány <u>különböző</u> előadó neve szerepel a fájlban?

Kiindulásképpen megint jó lesz nekünk a **cat** parancs, amivel kiírathatjuk a fájl tartalmát.

```
cservz@debian:~$ cat playlist.csv
Rick Astley;Never Gonna Give You Up;pop;213
Imagine Dragons;Thunder;pop;204
Dragonforce;Through the Fire and Flames;metal;445
Boney M.;Rasputin;pop;284
Steppenwolf;Born To Be Wild;rock;216
Powerwolf;Incense and Iron;metal;240
Smash Mouth;All Star;rock;237
Nirvana;Smells Like Teen Spirit;rock;279
Gloryhammer;The Unicorn Invasion of Dundee;metal;265
Powerwolf;We Are The Wild;metal;221
Imagine Dragons;Radioactive;rock;188
Dschinghis Khan;Moskau;pop;275
Dschinghis Khan;Dschinghis Khan;pop;185
Bonnie Tyler;Total Eclipse of the Heart;pop;334
Gopnik McBlyat;Snakes In Tracksuits;hardbass;261
John Farnham;Thunder In Your Heart;rock;234
```

Az előadók nevei minden sorban a pontosvesszővel elválasztott adatok közül a legelsők. A **cut** paranccsal feldarabolhatjuk a fájl sorait oszlopokra, és lekérhetjük egy adott oszlop tartalmát. A **-d** kapcsolóval megadjuk, hogy milyen karakterek mentén szeretnénk darabolni (ez most a pontosvessző lesz, hiszen a sorokban szereplő adatok pontosvesszővel vannak elválasztva), majd a **-f** kapcsolóval megmondjuk, hogy a feldarabolás után hányadik oszlopot szeretnénk lekérni (most az első oszlop fog kelleni, hiszen ez tartalmazza az előadók nevét).

A pipeline-unk eddig:

```
cservz@debian:~$ cat playlist.csv | cut -d ";" -f 1
Rick Astley
Imagine Dragons
Dragonforce
Boney M.
Steppenwolf
Powerwolf
Smash Mouth
Nirvana
Gloryhammer
Powerwolf
Imagine Dragons
Dschinghis Khan
Dschinghis Khan
Bonnie Tyler
Gopnik McBlyat
John Farnham
```

Nagyszerű, megvannak az előadók nevei. Látható viszont, hogy a parancs kimenetében néhány előadó neve többször is szerepel (pl. Imagine Dragons, Powerwolf, Dschinghis Khan). Mivel a feladat szövege a különböző előadók számára kíváncsi, ezért jó lenne ezeket az ismétlődéseket megszüntetnünk. Az első gondolatunk a **uniq** parancs lehet.

```
cservz@debian:~$ cat playlist.csv | cut -d ";" -f 1 | uniq
Rick Astley
Imagine Dragons
Dragonforce
Boney M.
Steppenwolf
Powerwolf
Smash Mouth
Nirvana
Gloryhammer
Powerwolf
Imagine Dragons
Dschinghis Khan
Bonnie Tyler
Gopnik McBlyat
John Farnham
```

Hát, ez közel sem lett tökéletes, hiszen az Imagine Dragons és Powerwolf előadónevek még mindig előfordulnak többször is. Egyedül a Dschinghis Khan ismétlődését sikerült megszüntetnünk a **uniq** használatával.

Miért van ez így? A **uniq**-ról fontos tudni, hogy csak az egymás utáni ismétlődéseket szünteti meg. Mivel a példafájlban az egyik Dschighis Khan közvetlenül a másik Dschinghis Khan után szerepelt, ezért ezzel a **uniq** el tudott bánni.

Hogyan lehetne vajon elérni, hogy ne csak az egymás utáni ismétlődéseket, hanem az összes ismétlődést megszüntessük? Erre a következő trükköt tudjuk használni: a **sort**-tal először ábécé sorrendbe rendezzük az előadók neveit (így az ismétlődő előadónevek garantáltan közvetlenül egymást fogják követni), majd csak ezután használjuk a **uniq**-ot.

```
cservz@debian:~$ cat playlist.csv | cut -d ";" -f 1 | sort | uniq
Boney M.
Bonnie Tyler
Dragonforce
Dschinghis Khan
Gloryhammer
Gopnik McBlyat
Imagine Dragons
John Farnham
Nirvana
Powerwolf
Rick Astley
Smash Mouth
Steppenwolf
```

Szuper, most már mindenki neve csak egyszer fordul elő. Már csak meg kell számolnunk, hogy hány darab adatot válogattunk ki (tehát az eddigi pipeline kimenete hány sorból áll). Erre a korábban tárgyalt **wc -l** parancsot használhatjuk. A feladat megoldása tehát:

```
cservz@debian:~$ cat playlist.csv | cut -d ";" -f 1 | sort | uniq | wc -l
13
```

3. Írasd ki a 10. sorban szereplő zeneszám címét!

Ha egy fájl bizonyos sorát (vagy sorait) szeretnénk csak kiíratni, akkor emlékezzünk vissza a **head** és **tail** parancsokra:

- **head -<szám> <fájl>:** kiírja a <fájl> első <szám> darab sorát
- **tail -<szám> <fájl>:** kiírja a <fájl> utolsó <szám> darab sorát.

Induljunk ki a **head -10 playlist.csv** parancsból! Ez kiírja a playlist.csv első 10 sorát.

```
cservz@debian:~$ head -10 playlist.csv
Rick Astley;Never Gonna Give You Up;pop;213
Imagine Dragons;Thunder;pop;204
Dragonforce;Through the Fire and Flames;metal;445
Boney M.;Rasputin;pop;284
Steppenwolf;Born To Be Wild;rock;216
Powerwolf;Incense and Iron;metal;240
Smash Mouth;All Star;rock;237
Nirvana;Smells Like Teen Spirit;rock;279
Gloryhammer;The Unicorn Invasion of Dundee;metal;265
Powerwolf;We Are The Wild;metal;221
```

Nekünk viszont most nem az első 10 sorra, hanem csupán a tizedik sorra van szükségünk. Egy kis gondolkodással rájöhetünk, hogy ha az első 10 sorból vesszük az utolsót, akkor pont a tizedik sort fogjuk eredményül megkapni. A **tail -1** parancs segítségével megkaphatjuk az előző parancs kimenetének az utolsó sorát.

```
cservz@debian:~$ head -10 playlist.csv | tail -1
Powerwolf;We Are The Wild;metal;221
```

Oké, a nehezén túlvagyunk, már csak az eredményül kapott sorból kellene lekérnünk a zene címét. Ez már viszonylag egyszerű: a **cut** paranccsal feldaraboljuk a sort pontosvesszők mentén, és vesszük a címet tartalmazó 2. oszlopot. A feladat megoldása tehát:

```
cservz@debian:~$ head -10 playlist.csv | tail -1 | cut -d ";" -f 2
We Are The Wild
```

4. Hány másodperc hosszú a fájlban található leghosszabb zeneszám?

Haladjunk megint kisebb lépésekben! Mivel ismét a fájlból szeretnénk kiíratni valamit, ezért kiindulásképpen kiírathatjuk a fájl tartalmát.

```
cservz@debian:~$ cat playlist.csv
Rick Astley;Never Gonna Give You Up;pop;213
Imagine Dragons;Thunder;pop;204
Dragonforce;Through the Fire and Flames;metal;445
Boney M.;Rasputin;pop;284
Steppenwolf;Born To Be Wild;rock;216
Powerwolf;Incense and Iron;metal;240
Smash Mouth;All Star;rock;237
Nirvana;Smells Like Teen Spirit;rock;279
Gloryhammer;The Unicorn Invasion of Dundee;metal;265
Powerwolf;We Are The Wild;metal;221
Imagine Dragons;Radioactive;rock;188
Dschinghis Khan;Moskau;pop;275
Dschinghis Khan;Dschinghis Khan;pop;185
Bonnie Tyler;Total Eclipse of the Heart;pop;334
Gopnik McBlyat;Snakes In Tracksuits;hardbass;261
John Farnham;Thunder In Your Heart;rock;234
```

Mivel a soroknak csak egy bizonyos része (a zene hossza) érdekel minket, ezért a **cut** paranccsal ismételten feldarabolhatjuk a sorok tartalmát pontosvesszők mentén. A zene hosszát a 4. oszlop tartalmazza, ezért ezt fogjuk lekérni.

```
cservz@debian:~$ cat playlist.csv | cut -d ";" -f 4
213
204
445
284
216
240
237
279
265
221
188
275
185
334
261
234
```

Oké, megkaptuk a zenék hosszát (másodpercben). Ahhoz, hogy meg tudjuk állapítani a leghosszabb zene hosszát, rendeznünk kell ezeket az értékeket. A **sort** parancsot fogjuk használni, viszont ne feledjük, hogy ha számokat szeretnénk rendezni, akkor használnunk kell a **-n** kapcsolót is (egyéb esetben ábécé sorrendben rendezne).

```
cservz@debian:~$ cat playlist.csv | cut -d ";" -f 4 | sort -n
185
188
204
213
216
221
234
237
240
261
265
275
279
284
334
445
```

Nagyszerű, most már növekvő sorrendben szerepel a zenék hossza. Már csak a leghosszabb zene hosszát kell lekérnünk, ami a növekvő sorrendben szereplő utolsó érték lesz. Az utolsó érték lekéréséhez használjuk a korábban tárgyalt **tail -1** parancsot! A megoldás tehát:

```
cservz@debian:~$ cat playlist.csv | cut -d ";" -f 4 | sort -n | tail -1
445
```

5. Hány pop zene van a fájlban?

A pop zenék meglepő módon azok a zenék, amelyeknek a műfaja `pop`. Tehát szükségünk van a `playlist.csv` fájl azon soraira, amelyek tartalmazzák a `pop` szöveget.

Ha egy fájl bizonyos sorait szeretnénk illeszteni egy megadott mintára, akkor az **`egrep`** parancsot kell használnunk. A parancs első paramétere a minta, amire illesztünk (amilyen szöveget keresünk a fájl soraiban), második paraméterben pedig megadjuk a fájl nevét.

```
cservz@debian:~$ egrep "pop" playlist.csv
Rick Astley;Never Gonna Give You Up;pop;213
Imagine Dragons;Thunder;pop;204
Boney M.;Rasputin;pop;284
Dschinghis Khan;Moskau;pop;275
Dschinghis Khan;Dschinghis Khan;pop;185
Bonnie Tyler;Total Eclipse of the Heart;pop;334
```

A parancs hatására már csak a pop zenék adatai kerültek kiíratásra. Már csak meg kell számolnunk, hogy hány zenét kaptunk eredményül. Ezt a szokásos módon, a **`wc -l`** paranccsal tehetjük meg. A feladat egy lehetséges megoldása tehát:

```
cservz@debian:~$ egrep "pop" playlist.csv | wc -l
6
```

Megjegyzés 1: Az **`egrep`** által visszaadott adatok megszámlálására a **`wc -l`** helyett használhatjuk az **`egrep`** parancs **`-c`** kapcsolóját is (ez kiírja a kiválogatott sorok számát). Tehát a megoldás lehetne ez is: **`egrep -c "pop" playlist.csv`** .

Megjegyzés 2: A szemfülesebbek észrevehették, hogy ez nem egy tökéletes megoldás, ugyanis ha egy zene címe tartalmazná a `pop` szöveget, de a műfaja mondjuk `rock` lenne, akkor hibásan kiválogatnánk ezt is az előző parancsokkal. Viszont ha a feladat szövege nem kéri, hogy készüljünk fel ilyen esetekre, akkor hallgatólagosan feltesszük, hogy a `pop` szöveg csak a műfajoknál fordulhat elő.

Persze aki ezzel nem elégszik meg, az írhat egy ennél jóval precízebb pipeline-t, például:

- **`egrep -c ";pop;" playlist.csv`** vagy
- **`cat playlist.csv | cut -d ";" -f 3 | egrep -c "pop"`**.

6. Hány másodperc hosszú a leghosszabb olyan zene, amelynek műfaja `rock`?

Ez a feladat nagyon hasonlít a 4. feladatra, csupán annyi a különbség, hogy nem az összes zene közül akarjuk megkeresni a leghosszabbat, hanem csak a rock zenék közül.

Először is válogassuk ki a rock zenéket! Az **`egrep`** parancs segítségével keressük meg az összes olyan sort a fájlból amely tartalmazza a `rock` szöveget!

```
cservz@debian:~$ egrep "rock" playlist.csv
Steppenwolf;Born To Be Wild;rock;216
Smash Mouth;All Star;rock;237
Nirvana;Smells Like Teen Spirit;rock;279
Imagine Dragons;Radioactive;rock;188
John Farnham;Thunder In Your Heart;rock;234
```

Rendben, most már kiválogattuk az összes rock zenét. Már csak ezek közül kellene meghatároznunk a leghosszabb zenének a hosszát. A 4. feladat alapján ezt a következőképpen tehetjük meg:

- lekérdezzük az egyes rock zenék hosszát, ami a pontosvesszőkkel elválasztott adatok közül a 4. lesz minden sorban (**`cut -d ";" -f 4`**)
- növekvő sorrendbe rendezzük a zenék hosszát (**`sort -n`**)
- vesszük a növekvő sorrendbe rendezett értékek közül az utolsót (**`tail -1`**).

A feladat megoldása tehát:

```
cservz@debian:~$ egrep "rock" playlist.csv | cut -d ";" -f 4 | sort -n | tail -1
279
```

7. Írasd ki egy `out.txt` fájlba azoknak a zenéknek a címét csupa nagybetűvel, amelyek tartalmazzák a `Thunder` vagy a `Heart` szöveget! (Itt a "vagy" a logikai megengedő VAGY-ot jelenti, tehát nekünk azok a címek is jók, amelyek mindkét szöveget tartalmazzák.)

Hmm… itt már elég sok mindent kell csinálni. Az egyszerűség kedvéért először oldjuk meg a feladatot úgy, hogy egyelőre még a fájlba írással nem foglalkozunk, csupán a pipeline-t próbáljuk meg megírni!

Mivel ismét csak a fájl bizonyos soraira szeretnénk szűrni, ezért továbbra is az **`egrep`** parancs lesz a barátunk. Ebben a feladatban viszont egynél több mintára is illesztünk (`Thunder`, `Heart`), ezért használnunk kell az **`egrep`**-nek a **`-e`** kapcsolóját ezek megadásakor.

```
cservz@debian:~$ egrep -e "Thunder" -e "Heart" playlist.csv
Imagine Dragons;Thunder;pop;204
Bonnie Tyler;Total Eclipse of the Heart;pop;334
John Farnham;Thunder In Your Heart;rock;234
```

Szuper, a kimenetből láthatjuk, hogy ténylegesen azoknak a zenéknek az adatait kaptuk meg, amelyek címe tartalmazza a `Thunder` és `Heart` szövegek legalább egyikét. A feladat szövege ezek közül az adatok közül csak a zenék címére kíváncsi. A zenecím a pontosvesszővel elválasztott adatok közül a második, így ezt a **`cut`** paranccsal egyszerűen lekérhetjük.

```
cservz@debian:~$ egrep -e "Thunder" -e "Heart" playlist.csv | cut -d ";" -f 2
Thunder
Total Eclipse of the Heart
Thunder In Your Heart
```

Oké, a következő feladatunk az lenne, hogy az így kapott kimenetet alakítsuk csupa nagybetűssé. Ezt a **`tr a-z A-Z`** paranccsal tehetjük meg, ami átalakítja a szövegben szereplő összes kisbetűt (`a-z`) nagybetűvé (`A-Z`).

```
cservz@debian:~$ egrep -e "Thunder" -e "Heart" playlist.csv | cut -d ";" -f 2 |
tr a-z A-Z
THUNDER
TOTAL ECLIPSE OF THE HEART
THUNDER IN YOUR HEART
```

Remek, most már a pipeline-nal készen vagyunk, kiírattuk a megfelelő zenék címét csupa nagybetűvel a konzolra. A feladat szövege viszont nem azt kérte, hogy a konzolra írassuk ki mindezt, hanem egy **out.txt** nevű fájlba kellene ezt beleraknunk.

A **>** operátorral egyszerűen átirányíthatjuk a pipeline kimenetét a megadott fájlba. Ha a fájl nem létezik, akkor ez a konstrukció automatikusan létrehozza azt, és beleírja a kimenetet.

```
cservz@debian:~$ egrep -e "Thunder" -e "Heart" playlist.csv | cut -d ";" -f 2 | tr a-z A-Z > out.txt
cservz@debian:~$ ls
pipeline_feladatok.pdf   playlist.csv   out.txt
cservz@debian:~$ cat out.txt
THUNDER
TOTAL ECLIPSE OF THE HEART
THUNDER IN YOUR HEART
```

Az out.txt fájlra vonatkozó jogosultságokat állítsd be a következők szerint:

- **a tulajdonosnak csak olvasási és írási joga legyen a fájlra**
- **a csoportnak szintén csak olvasási és írási joga legyen a fájlra**
- **a többiek ne rendelkezzenek semmilyen jogosultsággal a fájlra vonatkozóan!**

Ez a feladatrész tulajdonképpen már nem is a pipeline-okhoz kapcsolódik, hanem a jogosultságok kezeléséhez. Listázzuk ki az **out.txt** adatait részletesen!

```
cservz@debian:~$ ls -l out.txt
-rw-r--r-- 1 cservz cservz 57 Feb 5 12:23 out.txt
```

A listázás első oszlopában található `-rw-r--r--` szöveg fog minket érdekelni:

- az első kötőjel azt jelenti, hogy ez egy közönséges fájl (ha itt kötőjel helyett **d** szerepelne, akkor egy könyvtárral lenne dolgunk)
- a következő 3 karakter a fájl tulajdonosának a jogosultságait jelenti
- a következő 3 karakter a csoport jogosultságait jelenti
- az utolsó 3 karakter mindenki más jogosultságait írja le a fájlra vonatkozóan.

Az egyes felhasználói csoportoknak a jogosultságait rendre az **rwx** karakterek jelzik. Ezek sorban az olvasási (**r**ead), írási (**w**rite) és futtatási (e**x**ecute) jogokat jelölik a fájlra vonatkozóan. Ha valamelyik karakter helyén kötőjel szerepel, akkor az azt jelenti, hogy olyan jogosultsága nincs az adott felhasználói csoportnak az állományra.

A **chmod** paranccsal megváltoztathatjuk a fájlra vonatkozó jogosultságokat:

- először megadjuk, hogy kinek a jogosultságait szeretnénk módosítani: **u** (**u**ser, a tulajdonos), **g** (**g**roup, a csoport), **o** (**o**thers, a többiek), **a** (**a**ll, mindenki)
- ezután megmondjuk, hogy adni (**+**) vagy elvenni (**–**) szeretnénk jogosultságot
- végül megadjuk a jogosultságot jelölő karaktert (**r**, **w**, **x**).

A fájl tulajdonosának csak olvasási és írási jogot kell adnunk a feladat szövege szerint. Ha megnézzük az **ls -l** kimenetét, akkor ez már készen is van, hiszen a felhasználó jogosultságai: **rw-** (olvasási és írási joga van, de futtatási joga nincs).

A csoportnak szintén csak olvasási és írási joga kell, hogy legyen a fájlra vonatkozóan. A csoport jogosultságai: **r--**, tehát csak az olvasási jog van meg. Adjunk írási jogot (**w**) a csoportnak (**g**)!

```
cservz@debian:~$ chmod g+w out.txt
```

A többiek jogosultsága a fájlra vonatkozóan: **r–**, ami az olvasási jogosultságot jelenti. Mivel a feladat azt kérte, hogy a többiek ne rendelkeznek semmilyen jogosultsággal, ezért az olvasási jogot (**r**) el kell vennünk. Mivel jogot veszünk el, itt a mínuszjelet kell használnunk.

```
cservz@debian:~$ chmod o-w out.txt
```

A fájlra vonatkozó jogosultságok most már megfelelők:

```
cservz@debian:~$ ls -l out.txt
-rw-rw---- 1 cservz cservz 57 Feb 5 12:23 out.txt
```