Introductory Linux Tutorial for Life Sciences

Session 5: Practical examples & pair exercises

Linus Östberg

elixir SciLifeLab Data Centre

Today's schedule 🕒

$9^{00} - 10^{15}$	*Practical examples & pair exercises*
$10^{15} - 10^{30}$	**Coffee break**
$10^{30} - 12^{00}$	*Finding & compressing files*
$12^{00} - 13^{00}$	**Lunch**
$13^{00} - 15^{00}$	*Other util commands*
$15^{00} - 15^{15}$	**Coffee break**
$15^{15} - 17^{00}$	*Working remotely*

Shell commands

Filesystem	Text processing	Filters	Documentation	Editors	I/O redirection
cat	basename	grep	help	nano	>
cd	cut		man	vi	>>
cp	dirname				<
file	head				<<
ls	less				|
mv	sed				
mkdir	sort				
pwd	tail				
rm	tr				
rmdir	uniq				
touch	wc				
tree					

Command rehearsal and practical examples

1. Open the terminal and go from your home directory to the folder *myLinuxProject/results/courseDay_02* using either the absolute or the relative path. You might have to create the folder first.

```
$ mkdir -p /home/user/myLinuxProject/results/courseDay_02
$ cd /home/user/myLinuxProject/results/courseDay_02
$ cd myLinuxProject/results/courseDay_02
```

2. Copy the content of the file *~/data/others/myGroceryList.txt* to the new created file.

```
$ cp ~/data/others/myGroceryList.txt shoppingList.txt
```

Command rehearsal and practical examples

3. Visualize the content of the file *shoppingList.txt* using either the command less or cat:

```
$ cat shoppingList.txt

$ less shoppingList.txt
```

4. Sort the file *shoppingList.txt* numerically based on the third column in the list.

```
$ sort -nk 3 shoppingList.txt
```

Next, sort the file first by the name of the bought items and then numerically by their amount.

```
$ sort -k 2,2 -k 3n,3 shoppingList.txt
```

Command rehearsal and practical examples

5. The file *shoppingList.txt* contains 3 columns: the buyers name, the name of the bought items, and the amount. Extract the name of the bought items.

```
$ cut -d ' ' -f 2 shoppingList.txt
```

Sort the items to obtain those, which were bought several times:

```
$ cut -d ' ' -f 2 shoppingList.txt | sort | uniq -d
```

6. Change your list by replacing the *oranges* with *bananas*.

```
$ sed 's/oranges/bananas/' shoppingList.txt
```

Command rehearsal and practical examples

7. Identify all lines, which contain the pattern 'mellon'.

```
$ grep 'mellon' shoppingList.txt
```

Alter your command, so that it returns also the line numbers where the pattern is found.

```
$ grep -n 'mellon' shoppingList.txt
```

8. Find all words in the list with two or more vowels.

```
$ grep -E '[aeiou]{2}' shoppingList.txt
```

Command rehearsal and practical examples

9. Extract all lines, where the amount of bought items is two digit.

```
$ grep -E '[0-9].+' shoppingList.txt
```

10. Extract all lines that have the number 2 as the last character of the line

```
$ grep -E '2$' shoppingList.txt
```

11. Find all lines, which contain either 'oe' or 'ea'.

```
$ grep -E 'oe|ea' shoppingList.txt
```

Command rehearsal and practical examples

12. Reverse complement a fasta file

```
$ paste -d "\n" <(grep ">" myCDS.fa) <(grep -v ">" myCDS.fa|rev|tr "acgtACGT" "
    tgcaTGCA") >myCDS_revCompl.fa
```

1. extract the sequence identifier:

```
$ grep ">" myCDS.fa
```

2. reverse complement the sequences:

```
$ grep -v ">" myCDS.fa | rev | tr "acgtACGT" "tgcaTGCA"
```

3. combine the outcome from step 1) and 2) using the command paste

```
$ paste -d "\n" <(grep ">" myCDS.fa) <(grep -v ">" myCDS.fa|rev|tr "
    acgtACGT" "tgcaTGCA")
```

4. redirect the output to a file

```
$ paste -d "\n" <(grep ">" myCDS.fa) <(grep -v ">" myCDS.fa|rev|tr "
    acgtACGT" "tgcaTGCA") >myCDS_revCompl.fa
```

Command rehearsal and practical examples

13. Transform all lower case nucleotides to upper case nucleotides

```
$ paste -d "\n" <(grep ">" myCDS_revCompl.fa) <(grep -v ">" myCDS_revCompl.fa |
    tr "a-z" "A-Z")  >myCDS_revCompl_lc.fa
```

1. extract the sequence identifier

```
$ grep ">" myCDS_revCompl.fa
```

2. transform lower case nucleotides to upper case nucleotides

```
$ grep -v ">" myCDS_revCompl.fa | tr "a-z" "A-Z"
```

3. combine the outcome from step 1) and 2)

```
$ paste -d "\n" <(grep ">" myCDS_revCompl.fa) <(grep -v ">" myCDS_revCompl
    .fa | tr "a-z" "A-Z")
```

4. redirect the output to a new file

```
$ paste -d "\n" <(grep ">" myCDS_revCompl.fa) <(grep -v ">" myCDS_revCompl
    .fa | tr "a-z" "A-Z")  >myCDS_revCompl_lc.fa
```

Command rehearsal and practical examples

14. Extract sequences from the FASTA file *barley_CDS.fa* based on supplied identiers.

- extract the first 5 sequence identifiers from the given fasta file and save them in a new file:

```
$ grep ">" barley_CDS.fa | tr -d '>' | head -5 > myIds.txt
```

- use this identifiers to extract the corresponding sequences, too. This expects that the sequence information in the file *barley_CDS.fa* is represented in one line.

```
$ grep -A1 -F -f myIds.txt barley_CDS.fa | tr -d '-' > my_seq_ids.fa
```

grep -F – fixed strings

grep -A1 – print matching line plus the next line

Command rehearsal and practical examples

15. Determine all feature types annotated in a GFF file

```
grep -v "^#" ~/data/gff/Pygoscelis_adeliae.gff | cut -s -f 3 | sort | uniq
```

16. Determine the number of genes annotated in a GFF3 file

```
grep -c $'\tmRNA\t' Pygoscelis_adeliae.gff
```

17. Count the number of reads in a FASTQ file

```
$ cat ~/data/fastq/DRR001013.fastq | echo $((`wc -l`/4))
```

→ works only with 'back-ticks'

Command rehearsal and practical examples

18. Convert a FASTQ file to a FASTA file

```
$ sed -n '1~4s/^@/>/p; 2~4p' DRR001013.fastq > DRR001013.fasta
```

- M∼N with p – prints every N'th line starting with line first
- -n suppress automatic printing of pattern – will not print anything, unless an explicit request to print is found

Pair exercises - guidance

1. say hello to your next neighbor ☺→ he/she will be your partner for the next exercises
2. the remote participants will work alone → use the chat/video conference to communicate with each other
3. you need only the terminal integrated in the e-learning platform
4. solve the tasks by combining the commands taught yesterday
5. the exercises can be found on the e-learning platform under session 5

Pair exercises - guidance

1. say hello to your next neighbor ☺→ he/she will be your partner for the next exercises
2. the remote participants will work alone → use the chat/video conference to communicate with each other
3. you need only the terminal integrated in the e-learning platform
4. solve the tasks by combining the commands taught yesterday
5. the exercises can be found on the e-learning platform under session 5

There's more than one way to skin a cat!