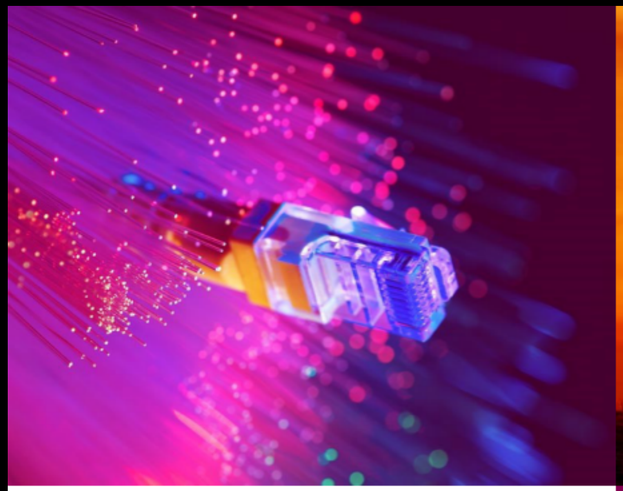

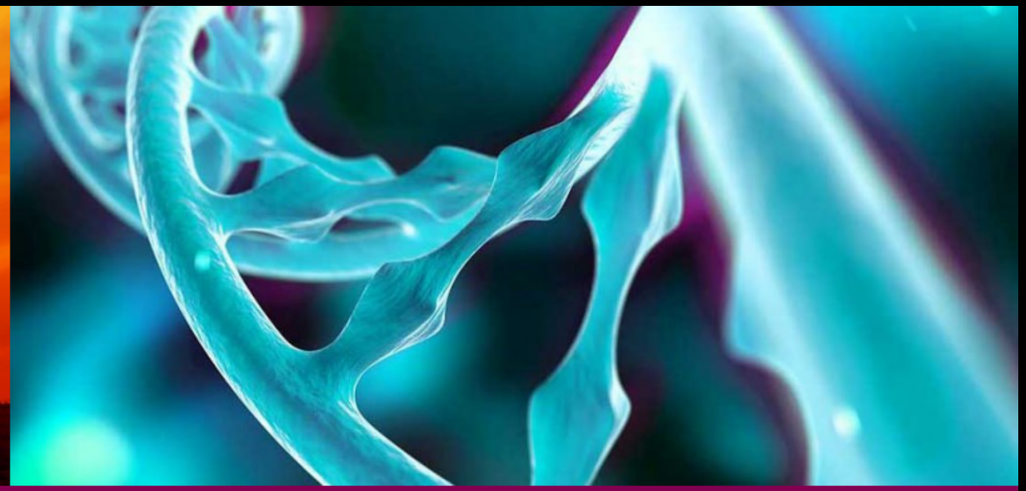

Replacing Spreadsheets - POSIX text utilities

CSC Training, 2019-12

CSC – Finnish expertise in ICT for research, education and public administration

What we will work over

- Your shell has built-in spread-sheet functions
- You can find/extract/combine text row or column-wise

Adding files side-by-side: paste

```
paste [-d del -s] file1 file2 [file3 …]
```

- Merges lines of several input files.
  - `-d` insert different delimiter *del* (between merged lines) than tabulator (default)
  - `-s` sequential in case of more than two files: `[(file1 + file2) + file3] + file4`
- Let’s try the following:

```
$ paste count.txt sheep.txt > counting_sheep_tab.txt          # creates merged file with tabulators
$ paste -d ' ' count.txt sheep.txt > counting_sheep_space.txt # creates merged file with spaces
```

Trimming files: cut

```
cut [-d del -f no -s] file1 file2 …
```

- Extracts fields/columns from each line of files.
  - `-d del` use different delimiter *del* (to identify fields) than tabulator (default)
  - `-f no` select fields *no*
  - `-s` skip lines not containing delimiters (e.g., header lines)
- Let’s try the following:

```
$ cut -f 1 counting_sheep_tab.txt
$ cut -f 1 -d ' ' counting_sheep_space.txt
```

- both will display the original content of count.txt

Counting lines [and sheep]: wc

```
wc [-l -w -m -c] file1 [file2 …]
```

- Counts lines, words as well as characters or bytes in a file (`wc` stands for **w**ord **c**ount):
  - `-l` count lines
  - `-w` count words
  - `-m` count characters
  - `-c` count bytes
  - without arguments displays lines, words, and byte-counts (as `-l -w -c`)
  - a word is a non-zero-length sequence of characters delimited by white space

```
$ wc -l sheep_space.txt
```

Combining files end to start: cat

```
cat [-n -E -v -T] file1 file2 …
```

- con**cat**enates files and prints to stdout.
  - `-n` numbering output lines (e.g., source-code listing)
  - `-E` indicate ends with a $
  - `-v` show non-printing
  - `-T` indicate tabs
- numbers the lines in `sheep_space.txt` and adds the column:

```
$ cat -n sheep_space.txt > sheep_lines.txt
```

```
$ cat -T -E sheep_tab.txt
```

Extracting beginning and end of files: head and tail

```
head [-n N] file1 [file2 …]
```

- Extracts head of files.
  - `-n N` display *N* first lines

```
tail [-n N -f --pid PID] file1 [file2 …]
```

- Extracts tail of files
  - `-n N` display *N* last lines
  - `-f` continuously display updates of file (useful to display log-files)
  - `--pid PID` terminate tail-command in sync with termination of process with process ID *PID*

Bringing order into files: sort

```
sort [-d -f -g ] file1 [file2 …]
```

- Sorts lines of text files (alphabetical or numerical).
  - `-d` dictionary (alphanumeric) order
  - `-f` ignore upper/lower case
  - `-g` general numeric
- Spot the difference:

```
$ sort -d sheep_space.txt
$ sort -g sheep_lines.txt
```

Removing redundancy in files: uniq

```
uniq [-c -f -s -w ] file1 [file2 …]
```

- Filters adjacent matching (redundant) files.
  - `-c` prefix lines by number of their occurrence
  - `-f N` avoid comparing the first *N* fields
  - `-s N` avoid comparing the first *N* characters
  - `-w N` compare not more than *N* characters/line
- Skips the first column (the previously inserted numbers) and matches in max. 10 characters (i.e., avoiding the later columns) and prefixes the number of occurrence (hint: try with –f 2):

```
$ uniq -c -f 1 -w 10 sheep_lines.txt
```