UNIX Programming Environment

command per input line

xargs cmd - run cmd on each line in STDIN

-P: process in parallel

Text Processing

Filtering with patterns

grep - print lines matching a pattern

-v: invert search result

-i: case insensitive

-n: line number of match

-R: recurse a structure

Regular expressions

. - match any character once

* - match previous pattern 0 or more times

+ - match previous pattern 1 or more times

[e-fF-M] - match any character in ASCII range F-M or e-f

[^e] - match all characters except e

^/$ - match beginning/end of line

() - group items together

| - match either left or right group

Simple text processing

tr - character translator

-s: replace repeating characters

-d: delete a character

cut - split a line into columns

Stream text processing

sed - modify string in various ways using pattern matching

Data Processing

Sorting

sort - (lexicographical) sorted concatenation

- **-r**: reverse the sort
- **-n**: numeric sort
- **-k**: merge by nth column
- **-t**: specifies separator character for -k

uniq - unique records in a sorted file

Joining

join - join lines of two sorted files on a common field

- **-1,-2**: specify fields in files that represent keys

Orchestrating pipelines

make - dependency-based command executor

- **-j**: execute in parallel

Task-based Tools

Remote host execution

ssh - securely login to remote server and get prompt; execute and capture

Contents from URLs

curl - query URL and print raw contents on terminal

- **-H**: set HTTP header
- **-i**: display headers received

Querying JSON data

json_pp - pretty-print JSON files

jq - Domain Specific Language (DSL) to query tree structure

Sync across hosts

rsync - sync files between directories

-a: archive mode, preserve permissions and access times

-v: files changed

--delete

Directory changes

inotifywait - watch directory for changes and print in a log

-m: monitor mode (run forever)

-r: watch directories recursively

bash

default command interpreter on UNIX environment

Variables

listing = 'ls -la' - initialisation

echo $listing - reference

$ export lgrep PATH - environment variable(declared by OS)

$ export PATH=$PATH:/home/yanchevk/bin - user modified

$ export lgrep PATH - all programs read as home/yanchevk/bin

Conditionals

```
if [-e 'test']; then
    echo "File exists"
else
    echo "File does not exist"
fi
```

[$foo = 'test'] - test string equality

[$num -eq 3] - test number equality

[! expression] - negate expression

Loops

```
for i in ‘seq 1 10’; do - iterate 1-10
    echo $i - print the number
done

for i in $(ls); do - iterate all files in directory
    echo ‘file --mime $i’
done

ls -fa ltr -s ‘ ‘lcut -f9 -d’ ‘l
while read file; do
    echo ‘file --mime $file’
done
```

Command line input

```
$0 - program name
$1 - first argument
$2 - second, etc.

argA=“defaultvalue”
while getopts “:a” opt; do
    case $opt in
        a)
            echo “-a was triggered!” >&2
            argA=$OPTARG
            ;;
        \?)
            echo “Invalid option: -$OPTARG” >&2
            ;;
    esac
done
```