Command line

A nice introduction to using Bash can be found here.

Some other sources include Bash Scripting Tutorial and the Linux Tutorial.

Some selected examples can be found in bash.pdf.

The `bash` interpreter

Bash is one of the most commonly used interpreters (languages) you will find on a typical *Linux* system.

When writing programs in *Bash* you will be able to use all the structures and functionality you expect from any other language, e.g. *Python*.)

Bash programs are scripts that start with a special first line.

```
#!/bin/bash
echo "Hello World"
```

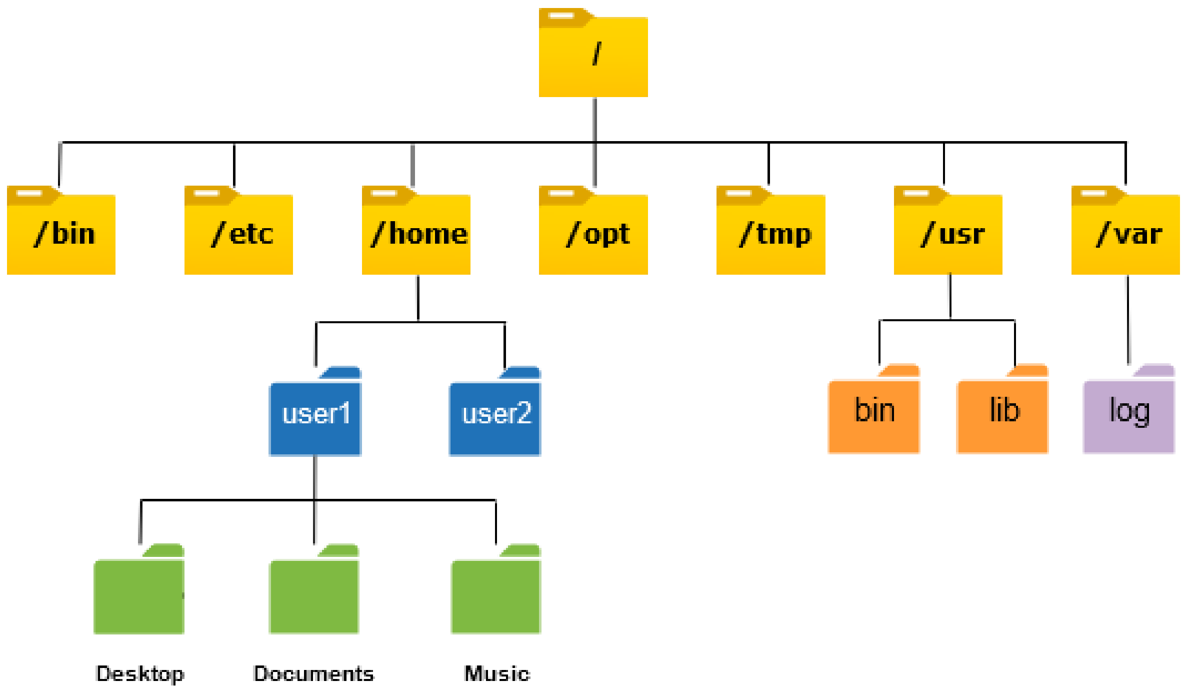
Store message into variable first.

```
#!/bin/bash
# declare STRING variable
STRING="Hello World"
#print variable on a screen
echo $STRING
```

The first line looks like it is commented out, because it starts with '#', but it is not. It instructs the command line interpreter to start the script in Bash (program `/bin/bash`).

1. Linux basics

Folder structure:



- files
  - ls
  - cp

  ```
  cp datoteke/primer.fasta kopija.fasta
  ```
  - rm
  - cat, less
  - gzip, zless

  ```
  gzip kopija.fasta # dobimo kopija.fasta.gz
      less kopija.fasta.gz
      zless kopija.fasta.gz
  ```
  - awk

  ```
  awk {'print $1 " text " $2'} datoteke/osebe.txt
  ```
- folders

  ```
  cd folder/path
      mkdir folder
      rmdir folder
  ```
- renaming and moving files

  ```
  mv from_name.txt to_name.txt
  ```
- programs
  - run a python script

2. Writing and running simple bash examples

- structure of a bash script
    - $ chmod +x simple.sh
- variables
    - global vs. local
- arithmetics
    - compute average weight in bash datoteke/teze.txt

```
#!/bin/bash
    echo "hello world"

    declare -i sum
    declare -i cn
    sum=0
    while read line; do
        sum=$sum+$line
        echo $line;
        cn=$cn+1
    done < datoteke/teze.txt

    echo "vsota: $sum"
    echo "n:" $cn
    echo "povprečje:" $sum/$cn
    echo "povprečje2:"  $[ $sum/$cn ]
    echo "povprečje3:" $(( $sum/$cn ))
```

- arguments
    - modify script to accept any input file
- running other programs within scripts
- user input
- if else
- comparisons
    - arithmetic
    - string
- loops – for iterating thru files, mostly
    - for
    - while

3. Pipelines

- Redirections of stdin, stdout, stderr to files
- Redirections with pipe
- Working on multiple files
  - Average of averages
  - Genome composition, one genome per file

Vaje

Napisi pythonov program, ki sprejme ime datotek in izracuna povprečje vrednosti.

average.py

```python
#!/srv/conda/envs/notebook/bin/python

import sys
fin_name = sys.argv[1]

sum = 0
cn = 0
for line in open(fin_name):
    sum += float(line)
    cn += 1

if cn > 0:
    print(sum/cn)
```

Napisi pythonov program, ki sprejme vsebino iz stdin in izracuna povprečje vrednosti.

average_stdin.py

```python
#!/srv/conda/envs/notebook/bin/python

import sys
fin = sys.stdin

sum = 0
cn = 0
for line in fin:
    sum += float(line)
    cn += 1

if cn > 0:
    print(sum/cn)
```

Program pričakuje novo vrednost (v novi vrstici) ali konec vnašanja (Ctrl+D).

napisi skripto v bash, ki pozene average.py na vsaki datoteki teze

pozeni_vse.sh

```bash
#!/bin/bash

for fn in datoteke/teze*.txt; do
    python average.py $fn
    # ./average.py $fn
    # cat $f | ./average.py
done
```

Napiši program, ki prebere vsako od datotek z zaporedji v datoteke/qwerty-dna/*.txt in za vsako izpiše sestavo a, t, c in g

prestej_dna.py

```python
#!/srv/conda/envs/notebook/bin/python
import sys

def prestej(s):
    frek = {}
    for c in s:
        frek[c] = frek.get(c, 0) + 1
    return frek

fin_name = sys.argv[1]
print(prestej(open(fin_name).read()))
```

Kličemo v programu prestej_vse.sh:

```bash
#!/bin/bash
for f in datoteke/qwerty-dna.txt; do
    echo "Procesiram $f"
    ./prestej_dna.py $f
done
```

Datoteko datoteke/qwerty-dna.txt rabij na posamezne datoteke.

razbij.sh

```bash
for f in `awk {'print $1'} datoteke/qwerty-dna.txt`; do
    grep "$f" datoteke/qwerty-dna.txt | awk {'print $2'} > $f.txt
done
```