DĚLAT
DOBRÝ SOFTWARE
NÁS BAVÍ

PROFINIT

Big data prerequisities

Sergii Stamenov

October 9, 2019

Osnova

- Základy linux
- Základy SQL
- Základy python

PROFINIT

LINUX

Linux

- Linux je rodina operačních systémů, na základě Linux kernel.
- Nejpopulárnější serverový operační systém. (Red Hat Linux anebo SUSE Linux.)
- Na PC mají největší popularitu Ubuntu, Debian a Fedora.

Unix filesystem

- Unix fs má stromovou strukturu. Umístění kořenu se označuje /. Všechno je soubor. Včetně I/O device.

```
ls -l /
```

```
(JESSIE)pascepet@hador:~$ ls -l /
total 105
drwxr-xr-x   2 root root        8192 Aug 14 08:23 bin
drwxr-xr-x   3 root root        4096 Aug 14 08:30 boot
drwxr-xr-t   3 hdfs hadoop        17 Oct 10  2016 data
drwxr-xr-x  20 root root        3940 Oct  6 06:25 dev
drwxr-xr-x 162 root root       12288 Oct  4 13:31 etc
lrwxrwxrwx   1 root root          19 Jan 20  2017 home -> /storage/brno2/home
drwxr-xr-x  18 root root        4096 Apr 26  2018 lib
drwxr-xr-x   2 root root        4096 Jun 20  2017 lib32
drwxr-xr-x   2 root root          83 Jun 20  2017 lib64
drwxr-xr-x   3 root root          17 Mar  3  2017 mnt
drwxr-xr-x   7 root root          99 Dec 14  2018 opt
dr-xr-xr-x 446 root root           0 Apr 26 10:10 proc
drwx------  22 root root        4096 Aug 16 14:12 root
drwxr-xr-x  37 root root        1400 Oct  6 10:00 run
drwxr-xr-x   2 root root        8192 Aug 14 08:24 sbin
dr-xr-xr-x  13 root root           0 Apr 26 10:12 sys
drwxrwxrwt 122 root root       12288 Oct  6 10:01 tmp
drwxr-xr-x  11 root root         109 Jun 27  2015 usr
drwxr-xr-x  12 root root         138 Sep 18  2017 var
```

PROFINIT

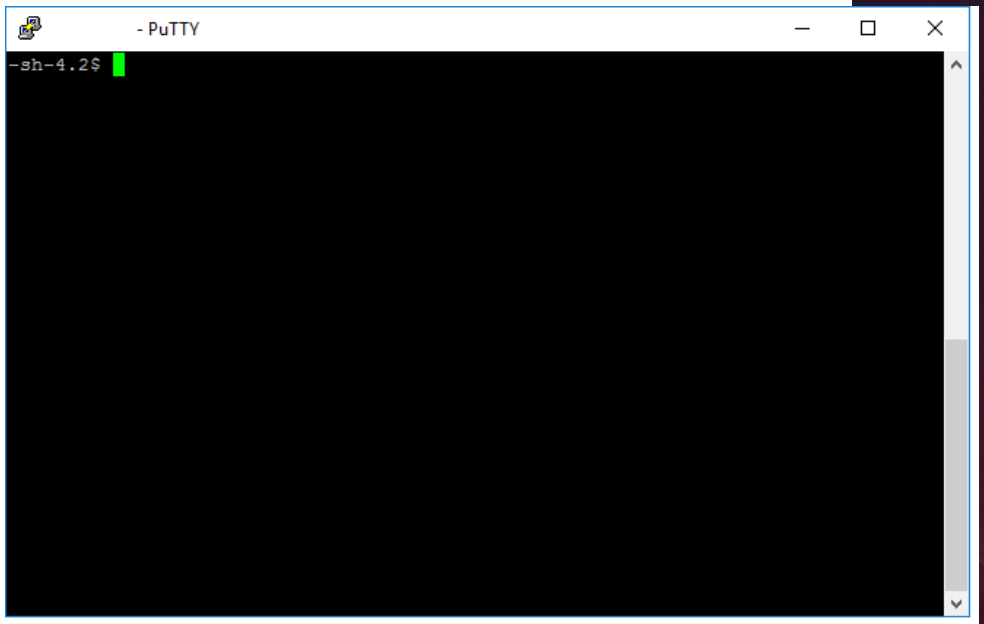
Několik důležitých složek

- **/bin**
  - programy/binárky které můžete spustit, ekvivalent .exe souboru ve windows
- **/dev**
  - aka device, driver zařízení
- **/etc**
  - konfigurační soubory
- **/lib**
  - sdílené knihovny
- **/home**
  - uživatelské adresáře, např. /home/stameser/
- **/proc**
  - všechny běžící procesy
- **/tmp**
  - dočasné soubory
- **/opt**
  - programy, které nejsou součásti standardní instalace
- **/var**
  - logy

Shell

- Interaktivní vývojářské rozhraní je příkazová řádka/terminal.

- Bourne shell
  - : Bourne shell (sh)
  - : Korn shell (ksh)
  - : Bourne again shell (bash)
  - : POSIX shell (sh)
- C shell
  - : C shell (csh)
  - : TENEX/TOPS C shell (tcsh)



- Mužete zkontrolovat, jaký máte shell, pomocí příkazu

```
(JESSIE)pascepet@hador:~$ echo $SHELL
/bin/bash
```

PROFINIT

ENV a PATH

- $SHELL - je odkaz na proměnnou prostředí. To jsou globální konstanty, které jsou k dispozici pro každý shell.
- Můžete definovat vlastní proměnné. MYVAR="BDT"

```
(JESSIE)pascepet@hador:~$ MYVAR=BDT
(JESSIE)pascepet@hador:~$ echo $MYVAR
BDT
```

- Proměnné existuji jenom v rámci sessiony. Po uzavření terminálu všechny proměnné zmizí. Pokud je chcete uložit natrvalo, musíte je zapsat do souboru .bash_profile.
- Pomocí příkazu **env** můžete vypsat všechny proměnné prostředí.

PROFINIT

PATH

- PATH – speciální proměnná, určuje adresáře, kde bude shell hledat definice příkazů. Obsahuje seznam cest oddělených dvojtečkou:

```
(JESSIE)pascepet@hador:~$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games:/software/meta-
utils/public
```

```
(JESSIE)pascepet@hador:~$ hello
-bash: hello: command not found
```

- Priorita cest je sestupná zleva doprava.
- Pomocí příkazu **`which`** můžeme najít cestu k příkazu.

```
(JESSIE)pascepet@hador:~$ which bash
/bin/bash
```

PROFINIT

Navigace

- Kde jsem?
  - **pwd** – print working directory (folder)
- Kdo jsem?
  - **whoami** – print current user
- Co tu mám?
  - **ls** [path] – list folder
  - **ll** – list folder with extended info
- Chci někam
  - **cd** [path] – change directory (folder)

```
(JESSIE)pascepet@hador:~$ pwd
/storage/brno2/home/pascepe
(JESSIE)pascepet@hador:~$ cd /opt
(JESSIE)pascepet@hador:/opt$ pwd
/opt
(JESSIE)pascepet@hador:/opt$ cd
#anebo cd ~/
(JESSIE)pascepet@hador:~$ pwd
/storage/brno2/home/pascepe
```

PROFINIT

Práce se soubory

- **cp** [src] [dst] - copy file
- **cp** -R [src] [dst] - copy folder and subfolders
- **mv** [src] [dst] - move/rename file or folder
- **rm** [file] - remove file
- **rm** -R [file] - remove folder

- **mkdir** – make (create) new folder

- **cat** [file] [file] - print file content
- **head/tail** [file] - print first/last rows of the file
- **less/more** [file] - browse through the file

PROFINIT

Práce se soubory – demo

```
(JESSIE)pascepet@hador:~/shell_demo$ ll
total 4
-rw-r--r-- 1 pascepet 4294967294 12 Oct  6 11:10 example.txt

(JESSIE)pascepet@hador:~/shell_demo$ mkdir new_folder

(JESSIE)pascepet@hador:~/shell_demo$ ll
total 4
-rw-r--r-- 1 pascepet 4294967294 12 Oct  6 11:10 example.txt
drwxr-xr-x 2 pascepet 4294967294  6 Oct  6 11:11 new_folder

(JESSIE)pascepet@hador:~/shell_demo$ cat example.txt
hello wolrd

(JESSIE)pascepet@hador:~/shell_demo$ cp example.txt hello_cp.txt
(JESSIE)pascepet@hador:~/shell_demo$ ll
total 8
-rw-r--r-- 1 pascepet 4294967294 12 Oct  6 11:10 example.txt
-rw-r--r-- 1 pascepet 4294967294 12 Oct  6 11:12 hello_cp.txt
drwxr-xr-x 2 pascepet 4294967294  6 Oct  6 11:11 new_folder

(JESSIE)pascepet@hador:~/shell_demo$ cat hello_cp.txt
hello wolrd
```

PROFINIT

Prace se soubory demo

```
(JESSIE)pascepet@hador:~/shell_demo$ rm example.txt
(JESSIE)pascepet@hador:~/shell_demo$ ll
total 4
-rw-r--r-- 1 pascepet 4294967294 12 Oct  6 11:12
hello_cp.txt
drwxr-xr-x 2 pascepet 4294967294  6 Oct  6 11:11
new_folder


(JESSIE)pascepet@hador:~/shell_demo$ mv hello_cp.txt
new_folder/example.txt
(JESSIE)pascepet@hador:~/shell_demo$ ll
total 0
drwxr-xr-x 2 pascepet 4294967294 24 Oct  6 11:14
new_folder
(JESSIE)pascepet@hador:~/shell_demo$ ll new_folder/
total 4
-rw-r--r-- 1 pascepet 4294967294 12 Oct  6 11:12
example.txt


(JESSIE)pascepet@hador:~/shell_demo$ rm -r
new_folder/
(JESSIE)pascepet@hador:~/shell_demo$ ll
total 0
```

PROFINIT

Přístupová práva

```
-rw-r--r-- 1 pascepet 4294967294 12 Oct  6 11:12 hello_cp.txt
drwxr-xr-x 2 pascepet 4294967294  6 Oct  6 11:11 new_folder
```

- Každý soubor má 3 nastavení přístupových práv.
  - Vlastník rw-
  - Grupa r--
  - Ostatní r—
- Druhy práv
  - r – read (4)
  - w – write (2)
  - x – execute (1)
- Změna nastavení proběhá pomocí příkazu **chmod**
  - **chmod** [mod] [file] - set rights on file
  - **chmod** -R [mod] [file] - set rights on folder and all subfolders

```
(JESSIE)pascepet@hador:~/shell_demo$ ll
total 4
-rw-r--r-- 1 pascepet 4294967294 12 Oct  6 11:22 example.txt
(JESSIE)pascepet@hador:~/shell_demo$ chmod 666 example.txt
(JESSIE)pascepet@hador:~/shell_demo$ ll
total 4
-rw-rw-rw- 1 pascepet 4294967294 12 Oct  6 11:22 example.txt
```

PROFINIT

Základy text processingu

- Jak zjistit počet slov/řádků v souboru?
  - **wc** [option] [file]

```
(JESSIE)pascepet@hador:~/shell_demo$ cat echo.log
hello wolrd
hello wolrd
hello wolrd 123

# pocet radku
(JESSIE)pascepet@hador:~/shell_demo$ wc echo.log -l
3 echo.log

# pocet slov
(JESSIE)pascepet@hador:~/shell_demo$ wc echo.log -w
7 echo.log
```

- Jak najít řádky, které (ne)obsahují řetězec?
  - **grep** [pattern] [file]

```
(JESSIE)pascepet@hador:~/shell_demo$ grep -P "\d{3}" echo.log
hello wolrd 123

(JESSIE)pascepet@hador:~/shell_demo$ grep -v -P "\d{3}" echo.log
hello wolrd
hello wolrd
```

PROFINIT

Přesměrovaní výstupu

- Když chci uložit výstup programu do souboru, můžu použít >

```
(JESSIE)pascepet@hador:~/shell_demo$ echo "hello wolrd" > echo.log
(JESSIE)pascepet@hador:~/shell_demo$ cat echo.log
hello wolrd

(JESSIE)pascepet@hador:~/shell_demo$ echo "hello wolrd" > echo.log
(JESSIE)pascepet@hador:~/shell_demo$ cat echo.log
hello wolrd

(JESSIE)pascepet@hador:~/shell_demo$ echo "hello wolrd" >> echo.log
(JESSIE)pascepet@hador:~/shell_demo$ cat echo.log
hello wolrd
hello wolrd
```

- Znak < umožňuje načíst obsah souboru na standardní vstup příkazu.

```
(JESSIE)pascepet@hador:~/shell_demo$ wc -w < echo.log
4
```

- Pipe čili zápis [cmda] | [cmdb] přesměrovává standardní výstup (stdout) příkazu A na standardní vstup příkazu B (stdin)

```
(JESSIE)pascepet@hador:~/shell_demo$ echo "Hello world" | wc -w
2
```

PROFINIT

SQL

Alfa a omega RDBMS

- Ve světě relačních databází je základní jednotkou informace **tabulka**. Tabulku lze chápat jako kolekce řádku s atributy (sloupečky).

- Jaké informace můžeme ukládat do sloupečku?
  - celočíselné int, bigint,
  - s plovoucí tečkou float / double
  - řetězce znaků string, char, varchar
  - datum date, datetime
  - ano/ne boolean
  - komplexní struktury array, map, struct

- **NULL** je speciální hodnota, označuje nevyplněný/chybějící údaj

PROFINIT

SQL DDL DML

- DDL – data definition language
  - Způsob, jak můžeme databázi říct, jak vypadají data.
  - CREATE DATABASE
  - CREATE TABLE
  - DROP TABLE
  - DROP DATABASE

- DML – data manipulation language
  - Manipulace s daty
  - SELECT
  - INSERT INTO
  - UPDATE (není v Hadoopu)
  - DELETE (není v Hadoopu)

PROFINIT

SELECT v kostce

```
SELECT [ALL | DISTINCT]
    select_expr,
    select_expr,
    ...
FROM table_reference
[WHERE where_condition]
[GROUP BY col_list]
[ORDER BY col_list]
[LIMIT [offset,] rows]
```

PROFINIT

Nejjednodušší databáze ever

```
use stameser;
show tables;

+-----------------------+--+
|       tab_name        |
+-----------------------+--+
| ap_temp               |
| ldap_group            |
+-----------------------+--+

create table employees (
    emp_id int,
    name string,
    dep_id int,
    salary int
);

create table departments (
    dep_id int,
    dep_name string
);

show tables;

+------------------------+--+
|        tab_name        |
+------------------------+--+
| ap_temp                |
| departments            |
| employees              |
| ldap_group             |
+------------------------+--+
```

Zaměstnanci

emp_id	name	dep_id	salary
1	Alice	1	6000
2	Bob	1	6000
3	Cris	null	7000
4	Marketa	2	5000
5	Brus	2	null

Oddělení

dep_id	name
1	development
2	marketing

PROFINIT

Vložení dat

```
describe employees;

+-----------+------------+----------+--+
| col_name  | data_type  | comment  |
+-----------+------------+----------+--+
| emp_id    | int        |          |
| name      | string     |          |
| dep_id    | int        |          |
| salary    | int        |          |
+-----------+------------+----------+--+

insert into departments (dep_id, dep_name) values (1, "dev"), (2, "marketing");

insert into employees values (1, "Alice", 1, 6000),
                             (2, "Bob", 1, 6000),
                             (3, "Cris", null, 7000),
                             (4, "Marketa", 2, 5000),
                             (5, "Brus", 2, null);


select * from employees limit 5;
+------------------+----------------+------------------+------------------+--+
| employees.emp_id | employees.name | employees.dep_id | employees.salary |
+------------------+----------------+------------------+------------------+--+
| 1                | Alice          | 1                | 6000             |
| 2                | Bob            | 1                | 6000             |
| 3                | Cris           | NULL             | 7000             |
| 4                | Marketa        | 2                | 5000             |
| 5                | Brus           | 2                | NULL             |
+------------------+----------------+------------------+------------------+--+
```

PROFINIT

Základní SELECT

- Projekce – výběr podmnožiny sloupečků

```
select name, salary from employees;
```

```
+----------+---------+--+
|   name   | salary  |
+----------+---------+--+
| Alice    | 6000    |
| Bob      | 6000    |
| Cris     | 7000    |
| Marketa  | 5000    |
| Brus     | NULL    |
+----------+---------+--+
```

- Filtrování – výběr podmnožiny řádků

```
select name, salary from employees where salary >= 6000;
```

```
+----------+---------+--+
|   name   | salary  |
+----------+---------+--+
| Alice    | 6000    |
| Bob      | 6000    |
| Cris     | 7000    |
+----------+---------+--+
```

PROFINIT

Základní SELECT

- Kontrola chybějících hodnot

```
select name, salary from employees where salary is not null;
+----------+---------+--+
|   name   | salary  |
+----------+---------+--+
| Marketa  | 5000    |
| Bob      | 6000    |
| Alice    | 6000    |
| Cris     | 7000    |
+----------+---------+--+
```

- Řazení

```
select name, salary from employees where salary is not null order by name desc;
+----------+---------+--+
|   name   | salary  |
+----------+---------+--+
| Marketa  | 5000    |
| Cris     | 7000    |
| Bob      | 6000    |
| Alice    | 6000    |
+----------+---------+--+
```

- asc – by default – vzestupně
- desc – sestupně

PROFINIT

Agregační SELECT

```
select
          count(*) as pocet_radku,
          count(salary) as pocet_hodnot,
          min(salary) as min_mzda,
          max(salary) as max_mzda,
          avg(salary) as prum_mzda
from employees;
```

```
+--------------+---------------+-----------+-----------+------------+--+
| pocet_radku  | pocet_hodnot  | min_mzda  | max_mzda  | prum_mzda  |
+--------------+---------------+-----------+-----------+------------+--+
| 5            | 4             | 5000      | 7000      | 6000.0     |
+--------------+---------------+-----------+-----------+------------+--+
```

PROFINIT

Agregační SELECT

- Agregace podle oddělení

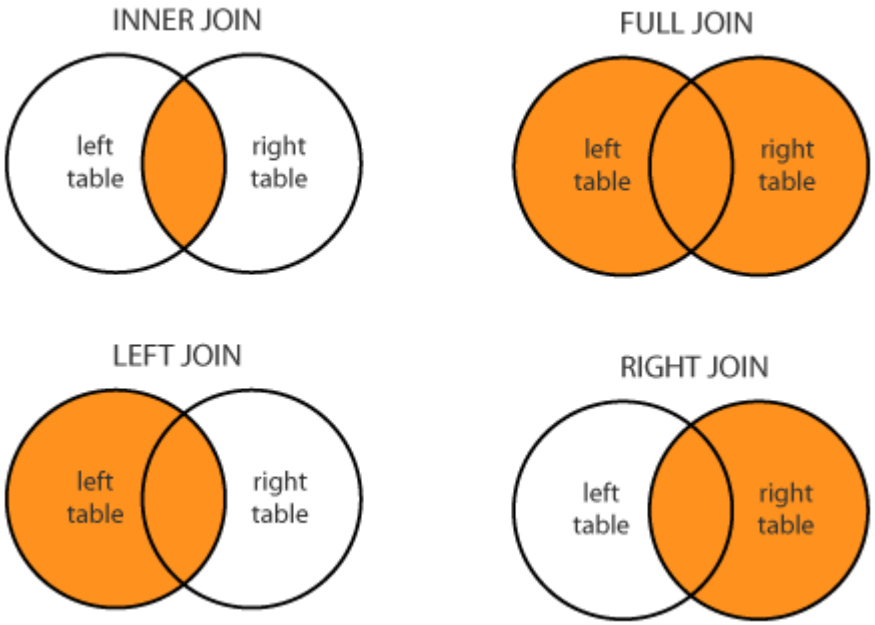
```
select dep_id,
           count(*) as pocet_radku,
           count(salary) pocet_hodnot,
           min(salary) as min_mzda,
           max(salary) as max_mzda,
           avg(salary) as prum_mzda
from employees
group by dep_id
order by pocet_radku;
```

```
+---------+--------------+---------------+-----------+-----------+------------+--+
| dep_id  | pocet_radku  | pocet_hodnot  | min_mzda  | max_mzda  | prum_mzda  |
+---------+--------------+---------------+-----------+-----------+------------+--+
| NULL    | 1            | 1             | 7000      | 7000      | 7000.0     |
| 2       | 2            | 1             | 5000      | 5000      | 5000.0     |
| 1       | 2            | 2             | 6000      | 6000      | 6000.0     |
+---------+--------------+---------------+-----------+-----------+------------+--+
```

PROFINIT

JOIN

- Propojení dvou tabulek podle klíče



JOIN příklad

Zaměstnanci

emp_id	name	dep_id	salary
1	Alice	1	6000
2	Bob	1	6000
3	Cris	null	7000
4	Marketa	2	5000
5	Brus	2	null

Oddělení

dep_id	name
1	development
2	marketing

```
select
    name,
    dep_name
from employees emp
join departments dep
   on emp.dep_id = dep.dep_id;
```

```
+----------+------------+--+
|   name   |  dep_name  |
+----------+------------+--+
| Alice    | dev        |
| Bob      | dev        |
| Marketa  | marketing  |
| Brus     | marketing  |
+----------+------------+--+
```

```
select
    name,
    dep_name
from employees emp
left join departments dep
   on emp.dep_id = dep.dep_id;
```

```
+----------+------------+--+
|   name   |  dep_name  |
+----------+------------+--+
| Alice    | dev        |
| Bob      | dev        |
| Cris     | NULL       |
| Marketa  | marketing  |
| Brus     | marketing  |
+----------+------------+--+
```

PROFINIT

PYTHON

PROFINIT

Děkujeme
za pozornost

PROFINIT

Profinit, s.r.o.
Tychonova 2, 160 00 Praha 6

Telefon
+ 420 224 316 016

Web
www.profinit.eu

LinkedIn
linkedin.com/company/profinit

Twitter
twitter.com/Profinit_EU