

# Podstawowy warsztat informatyka

PWI

Instytut Informatyki Uniwersytetu Wrocławskiego

## Wykład 3

*na podstawie slajdów Jakuba Michaliszyna*

## Dotychczas przerobiliśmy

- Operacje na plikach.
- Konta użytkowników.
- Łączenie z zasobami zdalnymi.

# Klucze prywatne i publiczne

```
$ ssh-keygen
```

```
Generating public/private rsa key pair.
```

```
Enter file in which to save the key (/home/me/.ssh/id_rsa):
```

```
Created directory '/home/me/.ssh'.
```

```
Enter passphrase (empty for no passphrase):
```

```
Enter same passphrase again:
```

```
Your identification has been saved in /home/me/.ssh/id_rsa.
```

```
Your public key has been saved in /home/me/.ssh/id_rsa.pub.
```

```
...
```

Klucz prywatny jest naszą tajemnicą!

Klucz publiczny wgrywamy na serwer:

```
$ ssh-copy-id www.example.com
```

i już!

# Prawa dostępu

- `ls -al`

```
drwxr-xr-x+ 1 jmi None      0 Oct  2 12:24 .
drwxrwxrwt+ 1 jmi None      0 Jan 23  2014 ..
-rw-----  1 jmi None 11531 Oct  7 17:05 .bash_history
-rwxr-xr-x  1 jmi None  1494 Jan 23  2014 .bash_profile
```

# Prawa dostępu

- `ls -al`  
drwxr-xr-x+ 1 jmi None 0 Oct 2 12:24 .  
drwxrwxrwt+ 1 jmi None 0 Jan 23 2014 ..  
-rw----- 1 jmi None 11531 Oct 7 17:05 .bash\_history  
-rwxr-xr-x 1 jmi None 1494 Jan 23 2014 .bash\_profile
- `d | rwx | rwx | rwx`  
czy katalog? | prawa właściciela | prawa grupy | prawa pozostałych

# Prawa dostępu

- `ls -al`  
drwxr-xr-x+ 1 jmi None 0 Oct 2 12:24 .  
drwxrwxrwt+ 1 jmi None 0 Jan 23 2014 ..  
-rw----- 1 jmi None 11531 Oct 7 17:05 .bash\_history  
-rwxr-xr-x 1 jmi None 1494 Jan 23 2014 .bash\_profile
- `d | rwx | rwx | rwx`  
czy katalog? | prawa właściciela | prawa grupy | prawa pozostałych
- `chmod; r=4, w=2, x=1.`

# Prawa dostępu

- `ls -al`  
drwxr-xr-x+ 1 jmi None 0 Oct 2 12:24 .  
drwxrwxrwt+ 1 jmi None 0 Jan 23 2014 ..  
-rw----- 1 jmi None 11531 Oct 7 17:05 .bash\_history  
-rwxr-xr-x 1 jmi None 1494 Jan 23 2014 .bash\_profile
- `d | rwx | rwx | rwx`  
czy katalog? | prawa właściciela | prawa grupy | prawa pozostałych
- `chmod; r=4, w=2, x=1.`
- `chmod +x` aby uczynić plik wykonywalnym, `./program` aby uruchomić program.

# Procesy

- `top`, `ps -f`



# Procesy

- `top`, `ps -f`
- Drzewiasta struktura - `pstree`.

# Procesy

- `top`, `ps -f`
- Drzewiasta struktura - `ps tree`.
- `/proc/`

# Procesy

- `top`, `ps -f`
- Drzewiasta struktura - `ps tree`.
- `/proc/`
- `chmod +x plik`  
  `./plik`

# Procesy

- `top`, `ps -f`
- Drzewiasta struktura - `pstree`.
- `/proc/`
- `chmod +x plik`  
  `./plik`
- `ctrl+c`, `ctrl+z`, `fg`, `bg`, `jobs`

# Zabijanie

- kill (-term, -kill, -l)

# Zabijanie

- kill (-term, -kill, -l)
- killall

# Interpreter poleceń

- Przekierowania: `cat /proc/cpuinfo > dane`  
`wc -l < dane`  
`rm > log`  
`wc 2> dane`

Dekryptor 0 (stdin) - domyślnie klawiatura, 1 (stdout) - ekran, 2 (stderr) - ekran.

# Interpreter poleceń

- Przekierowania: `cat /proc/cpuinfo > dane`

`wc -l < dane`

`rm > log`

`wc 2> dane`

Dekryptor 0 (stdin) - domyślnie klawiatura, 1 (stdout) - ekran, 2 (stderr) - ekran.

- Potoki: `cat /proc/cpuinfo | tee dane | wc -l .`



# Interpreter poleceń

- Przekierowania: `cat /proc/cpuinfo > dane`

`wc -l < dane`

`rm > log`

`wc 2> dane`

Dekryptor 0 (stdin) - domyślnie klawiatura, 1 (stdout) - ekran, 2 (stderr) - ekran.

- Potoki: `cat /proc/cpuinfo | tee dane | wc -l .`
- Równoczesne wykonanie! (cf. `cat | grep b`).

# PATH

- Zmienne:

$Y=Is$

$Y$

$\$Y$

# PATH

- Zmienne:  
Y=Is  
Y  
\$Y
- Zmienna PATH.

# PATH

- Zmienne:  
Y=ls  
Y  
\$Y
- Zmienna PATH.
- `export PATH=$PATH: /opt/bin`

# PATH

- Zmienne:  
Y=ls  
Y  
\$Y
- Zmienna PATH.
- `export PATH=$PATH: /opt/bin`
- `printenv`

# PATH

- Zmienne:  
Y=ls  
Y  
\$Y
- Zmienna PATH.
- `export PATH=$PATH: /opt/bin`
- `printenv`
- `whereis`

## Co dalej?

- 24-30 października: AMPPZ - lista 4, do wysłania przez skos