

Linux Lab

Christian Kniep

Internation Center of Applied Technologies Bandung

4. August 2010

Table of content

- 1 Introduction
 - get to know the terminal
- 2 Listings & Permissions
 - ls-Command
 - Types
 - Permissions
- 3 Doing
 - Basic commands
 - practical
 - additional

log in a remote computer

- First you have to log on the remote computer

```
$ ssh user@192.168.3.221
```

```
user@192.168.3.221's password:
```

log in a remote computer

- First you have to log on the remote computer

```
$ ssh user@192.168.3.221
```

```
user@192.168.3.221's password:
```

- Your usernames:

aegedio	maria
ryan	sintikhe
christian	yoseph

get to know the terminal

look around

● who am I `whoami`

get to know the terminal

look around

- who am I `whoami`
- where am I `pwd` **PathWorkingDirectory**

get to know the terminal

look around

- who am I `whoami`
- where am I `pwd` **PathWorkingDirectory**
- List your home-directory `ls`

get to know the terminal

look around

- who am I `whoami`
- where am I `pwd` **PathWorkingDirectory**
- List your home-directory `ls`
- OK, its empty. So create a directory `mkdir fstDir`

look around

- who am I `whoami`
- where am I `pwd` **PathWorkingDirectory**
- List your home-directory `ls`
- OK, its empty. So create a directory `mkdir fstDir`
- Whats the difference between
 - ▶ `ls`
 - ▶ `ls -l`
 - ▶ `ls -la`

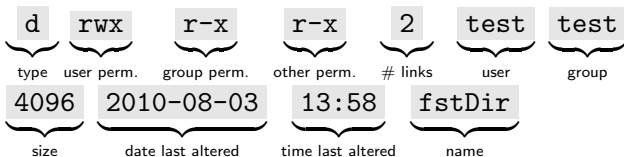
what does it mean?

- `drwxr-xr-x 2 test test 4096 2010-08-03 13:58 fstDir`

what does it mean?

- `drwxr-xr-x 2 test test 4096 2010-08-03 13:58 fstDir`

- what are these informations all about?



a few types you might encounter

- Basic types are:

short	description
-	regular file
d	directory
l	symbolic link (reference)

rwX?

- Basic permissions are:

short	long	file-context	dir-context
r	read	view the file	the dir is shown in <code>ls</code>
w	write	alter the file	create files/dir
x	execute	execute a file	enter the directory

rwX?

- Basic permissions are:

short	long	file-context	dir-context
r	read	view the file	the dir is shown in <code>ls</code>
w	write	alter the file	create files/dir
x	execute	execute a file	enter the directory

- Change the permissions

To change the permission use `chmod`

rwX?

- Basic permissions are:

short	long	file-context	dir-context
r	read	view the file	the dir is shown in <code>ls</code>
w	write	alter the file	create files/dir
x	execute	execute a file	enter the directory

- Change the permissions

To change the permission use `chmod`

- ▶ User: `chmod u=rwx file` or `u-r` or `u+x`

rwX?

- Basic permissions are:

short	long	file-context	dir-context
r	read	view the file	the dir is shown in <code>ls</code>
w	write	alter the file	create files/dir
x	execute	execute a file	enter the directory

- Change the permissions

To change the permission use `chmod`

- ▶ User: `chmod u=rwx file` or `u-r` or `u+x`
- ▶ Group / other: `g=` / `o=`

rwx?

- Basic permissions are:

short	long	file-context	dir-context
r	read	view the file	the dir is shown in <code>ls</code>
w	write	alter the file	create files/dir
x	execute	execute a file	enter the directory

- Change the permissions

To change the permission use `chmod`

- ▶ User: `chmod u=rwx file` or `u-r` or `u+x`
- ▶ Group / other: `g=` / `o=`

- Change the group: `chgrp <group>file`

cd, move and remove

- Change the **D**irectory `cd destination`
- move (equal to rename) is `mv file file1`
- remove (equal to rename) is `rm file`

create, read and write

- the simplest way to create a file is `touch file`

create, read and write

- the simplest way to create a file is `touch file`
- if you want to edit it you use `vim file`
 - ▶ now you are in the neutral mode

create, read and write

- the simplest way to create a file is `touch file`
- if you want to edit it you use `vim file`
 - ▶ now you are in the neutral mode
 - ▶ to edit the file type `i`, now there has to be the string `-- INSERT --` on the left bottom
 - ▶ now insert some stuff into your file

create, read and write

- the simplest way to create a file is `touch file`
- if you want to edit it you use `vim file`
 - ▶ now you are in the neutral mode
 - ▶ to edit the file type `i`, now there has to be the string `-- INSERT --` on the left buttom
 - ▶ now insert some stuff into your file
 - ▶ if you want to undo the last insertion go to the neutral mode (type `ESC` until the left buttom is clean) and type `u`

create, read and write

- the simplest way to create a file is `touch file`
- if you want to edit it you use `vim file`
 - ▶ now you are in the neutral mode
 - ▶ to edit the file type `i`, now there has to be the string `-- INSERT --` on the left buttom
 - ▶ now insert some stuff into your file
 - ▶ if you want to undo the last insertion go to the neutral mode (type `ESC` until the left buttom is clean) and type `u`
 - ▶ to write the file change to the neutral mode and type `:w`

create, read and write

- the simplest way to create a file is `touch file`
- if you want to edit it you use `vim file`
 - ▶ now you are in the neutral mode
 - ▶ to edit the file type `i`, now there has to be the string `-- INSERT --` on the left buttom
 - ▶ now insert some stuff into your file
 - ▶ if you want to undo the last insertion go to the neutral mode (type `ESC` until the left buttom is clean) and type `u`
 - ▶ to write the file change to the neutral mode and type `:w`
 - ▶ quitting is the command `:q`
 - ▶ its possible to do `:wq`

lets play

- change to `/var/linuxLab`

lets play

- change to `/var/linuxLab`
- have a look around, what do you see

lets play

- change to `/var/linuxLab`
- have a look around, what do you see
- try this in the current and in the subdirs

lets play

- change to `/var/linuxLab`
- have a look around, what do you see
- try this in the current and in the subdirs
 - ① create a file named as your user
 - ② insert your full name to it (use vim)

lets play

- change to `/var/linuxLab`
- have a look around, what do you see
- try this in the current and in the subdirs
 - ① create a file named as your user
 - ② insert your full name to it (use vim)
 - ③ add 1 to the name (rename it to `<username>1`)

lets play

- change to `/var/linuxLab`
- have a look around, what do you see
- try this in the current and in the subdirs
 - ① create a file named as your user
 - ② insert your full name to it (use vim)
 - ③ add 1 to the name (rename it to `<username>1`)
 - ④ add your name into your neighbours file

schedule

- lets pin down our leactures during the week
 - ① create a folder as your username `/var/linuxLab/<user>`

schedule

- lets pin down our leactures during the week
 - ① create a folder as your username `/var/linuxLab/<user>`
 - ② create a folder for each day of the week

schedule

- lets pin down our leactures during the week
 - ① create a folder as your username `/var/linuxLab/<user>`
 - ② create a folder for each day of the week
 - ③ create a file which contains the leactures and times in the day-folders

lazy linux

- help! I need somebody!

lazy linux

- help! I need somebody!
 - 1 The parameter `--help` give you a short explanation

lazy linux

- help! I need somebody!
 - 1 The parameter `--help` give you a short explanation
 - 2 `man <command>` gives you the full information

lazy linux

- help! I need somebody!
 - 1 The parameter `--help` give you a short explanation
 - 2 `man <command>` gives you the full information
- commands & parameters

lazy linux

- help! I need somebody!
 - 1 The parameter `--help` give you a short explanation
 - 2 `man <command>` gives you the full information
- commands & parameters
 - 1 if you type `!!` you got the last command

lazy linux

- help! I need somebody!
 - ① The parameter `--help` give you a short explanation
 - ② `man <command>` gives you the full information
- commands & parameters
 - ① if you type `!!` you got the last command
 - ② if you type `!$` you got the last parameter (really the last)

lazy linux

- help! I need somebody!
 - ① The parameter `--help` give you a short explanation
 - ② `man <command>` gives you the full information
- commands & parameters
 - ① if you type `!!` you got the last command
 - ② if you type `!$` you got the last parameter (really the last)
- cd

lazy linux

- help! I need somebody!
 - 1 The parameter `--help` give you a short explanation
 - 2 `man <command>` gives you the full information
- commands & parameters
 - 1 if you type `!!` you got the last command
 - 2 if you type `!$` you got the last parameter (really the last)
- cd
 - 1 type `cd` to change to your home

lazy linux

- help! I need somebody!
 - 1 The parameter `--help` give you a short explanation
 - 2 `man <command>` gives you the full information
- commands & parameters
 - 1 if you type `!!` you got the last command
 - 2 if you type `!$` you got the last parameter (really the last)
- cd
 - 1 type `cd` to change to your home
 - 2 type `cd -` to change to the previous path you where in

output, pipes

- if you fire a command the output will go through the Terminal

output, pipes

- if you fire a command the output will go through the Terminal
- dump it
 - ▶ a new file `ls -l >list`

output, pipes

- if you fire a command the output will go through the Terminal
- dump it
 - ▶ a new file `ls -l >list`
 - ▶ append it to a file `ls -l >>list`

output, pipes

- if you fire a command the output will go through the Terminal
- dump it
 - ▶ a new file `ls -l >list`
 - ▶ append it to a file `ls -l >>list`
- if you want to process it, pipe it in another process `ls -l |`
 - ▶ search for line with pattern `ls -l |grep pattern`

output, pipes

- if you fire a command the output will go through the Terminal
- dump it
 - ▶ a new file `ls -l >list`
 - ▶ append it to a file `ls -l >>list`
- if you want to process it, pipe it in another process `ls -l |`
 - ▶ search for line with pattern `ls -l |grep pattern`
 - ▶ the first two lines of the output `ls -l |head -n2`

output, pipes

- if you fire a command the output will go through the Terminal
- dump it
 - ▶ a new file `ls -l >list`
 - ▶ append it to a file `ls -l >>list`
- if you want to process it, pipe it in another process `ls -l |`
 - ▶ search for line with pattern `ls -l |grep pattern`
 - ▶ the first two lines of the output `ls -l |head -n2`
 - ▶ the last two lines of the output

output, pipes

- if you fire a command the output will go through the Terminal
- dump it
 - ▶ a new file `ls -l >list`
 - ▶ append it to a file `ls -l >>list`
- if you want to process it, pipe it in another process `ls -l |`
 - ▶ search for line with pattern `ls -l |grep pattern`
 - ▶ the first two lines of the output `ls -l |head -n2`
 - ▶ the last two lines of the output
 - ▶ `ls -l |tail -n2`

environment

- Where are all the commands located?

environment

- Where are all the commands located?

⇒ depends :) Mostly `/bin`, `/usr/bin`, `/usr/local/bin`

⇒ Environment. Try `echo $PATH`

environment

- Where are all the commands located?

⇒ depends :) Mostly `/bin`, `/usr/bin`, `/usr/local/bin`

- But how does Linux know where to look?

⇒ Environment. Try `echo $PATH`